2015

Towards an interactive environment for the performance of Dubstep music

James Herrington
Edith Cowan University

Link to videos of performances:

Performance 1: Instrumentalist-Controlled Interactive Dubstep

Performance 2: Dancer-Controlled Interactive Dubstep

Performance 3: Audience-Controlled Interactive Dubstep

Drums Practice

Synth Practice

Violin Stab Practice

Please contact Research Online for a downloadable copy of these videos.

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Towards an interactive environment for the performance of Dubstep music

James Herrington

Master of Arts (Performing Arts)
Music Composition and Technology

Western Australian Academy of the Performing Arts
Edith Cowan University

January, 2015

Abstract

This Masters by Research project explores the integration of different concepts relating to the presence of the human body in Dubstep music performance. Three intended performance systems propose that the body is the logical site for the interactive control of live Dubstep music. The physicality and gestures of instrumentalists, choreographed dancers, and audience members will be examined in order to develop new and exciting ways to perform this genre in a live setting.

The systems take on a three-tiered hierarchical approach on two levels in regards to the extraction of gestural information from human body movements, as well as in regards to the importance – and length – of musical phenomena and parameters that are under control. The characteristics of Dubstep music are defined and maintained within each interactive music system. A model for this each proposed system will be examined, including discussion of the technology and methodology employed in order to apply the two hierarchies and create the interactive environment.

Declaration

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education;

(ii) contain any material previously published or written by another person except where due reference is made in the text; or

(iii) contain any defamatory material.

I also grant permission for the Library at Edith Cowan University to make duplicate copies of my thesis as required.

Signed:

Date:

Acknowledgements

I wish to thank my supervisor, Lindsay Vickery, for the support and guidance he has shown me in the completion of this thesis, including in the writing of my dissertation and also in the execution of my practical research project. I appreciate very much his professional approach, his good humour, and his dedication to the role of supervisor.

I also wish to extend a special word of thanks to each and every participant involved in the three performances:

> Katie Campbell and Tim Cluett of Performance 1 – gifted musicians as well as good friends – they made the collaboration process as enjoyable as possible.
>
> Ezgi Gungor, Emily King and Melissa Tan of Performance 2 – to work with such passionate, enthusiastic and ambitious artists, who always maintained a fun-loving but professional perspective, provided a great experience.
>
> Finally, to everyone who partook in Performance 3 by donning a coloured t-shirt and dancing – without them, the night (and project) would not have been a success.

Finally, I wish to thank my Mum and Dad, for their continued support of my musical endeavors...even if they haven't always been the biggest fans of what I was creating.

Table of contents

List of figures & tables

Figures

Tables

CHAPTER 1

Introduction

Since the practice of performing electronic computer music – and especially the use of laptops in performance – became ubiquitous in the late 1990s (Stuart, 2003), concerns have been raised regarding the lack of human presence. The physicality of the human body in musical performance is seen to be a fundamental missing element (Turner, 2003), with some critics going as far as suggesting that the creating of successful art while 'performing' with a laptop is impossible (Ostertag, 2002). As they watch and listen on, it is hard for audiences to connect the physical gestures that they see with the musical outcomes that they hear (Stuart, 2003). There is also the lack-of-virtuosity factor, due to the fact that, as audiences cannot see the computer's desktop and what the artist is doing, they may in fact assume the person hiding behind the laptop is simply playing back musical excerpts on iTunes while checking their email (Collins, 2003). Even to audiences more adapted to the idea of laptops in musical performances, there is still a danger that they will underestimate the skill of the artist and complexity of the performance, or as Nick Collins puts it, the audience "understands that some music-making program is underway, but assume the use of a playback sequencer like Logic when the real performance is a live intergalactic jam with a being from the planet Sirius" (Collins, 2003, p. 68).

How to resolve this issue of performance in electronic computer music has been questioned, discussed and debated, however, it has yet to be decided. There are suggestions that 'playing' the laptop is a valid way to perform, and that performance expectations in fact undermine the listening experience (Cascone, 2003). In the same vein, it is also argued that changes in the way audiences understand the performances are required, from a visual focus to that of aural performativity (Stuart, 2003). In contrast to this way of thinking, however, is the idea that developments in musical interactivity are necessary (Chadabe, 2000; Keith, 2010), and that the visual stimulus of the human body needs to be brought back into the performance of electronic computer music (Ostertag, 2002).

Bob Ostertag considers the absence of the artist's body in electronic music performance a missing component necessary to the success of any work of art. He claims that:

> The problem was and still is how to get one's body into the unorthodox kind of performance we are talking about…With the emergence of the laptop as instrument, the physical aspect of the performance has been further reduced to sitting on stage and moving a cursor by dragging one's finger across a track pad in millimeter increments. (Ostertag, 2002, p. 12)

There are also perceptions of effortlessness stemming from the idea that "in electronic music the gestures of the performer usually do not seem commensurate to the sonic output that is being produced" (d'Escrivan, 2006, p. 183). Caleb Stuart also subscribes to this idea of thought, and claims that

> Performativity is directly linked to the body. We talk about the performance of the everyday in gender, sexuality, race and culture. In the performing arts and music, the body is generally near the action, on display. The objection is that, with laptop music, the audience is not being given the visual stimulus of a body that they are used to. That is, the body of the musician is not directly and causally, in a one-to-one relationship, acting on an object physically to create a sound. (Stuart, 2003, p. 64)

The aforementioned criticisms are especially pertinent to EDM, as the type of music is solely intended for the purpose of enticing listeners to dance, so there is an inherent sense of irony, as the on-stage performer/s' movements are often stagnant. Ableton Live has become a very popular computer software program used by producers and performers (or Disk Jockeys) of EDM (Keith, 2010), and therefore it is very common to see a DJ with a laptop on stage. Controllers such as digital turntables and MIDI keyboards are more often than not added to the equation, however, as the performers slide faders, twist knobs and press buttons – all hand-based gestures – the same criticisms of humanisation (or rather, the lack of) are raised (Ostertag, 2002).

Acclaimed EDM producer *Deadmau5* published a scathing examination of current EDM performance practice in a blog on Tumblr entitled *All DJs Are Glorified Button Pushers* ("Gizmodo", nd). His argument that a scene bereft of true live performance centres on the idea that significant pre-constructed blocks (or the arrangement of many set musical aspects) of the music are being triggered purely by limited actions; in this case, pushing buttons, and then occasionally "twiddlin a knob or somethin" to merely tweak the pre-mixed audio. His own performance style is not spared of the criticism, as he goes on to explain how his set-up "doesn't give (him) alot of 'lookit me I'm Jimi Hendrix check out this solo' stuff" ("Gizmodo", nd).

In his article *Disclosure Unplugged: Why DJs Need To Step Down From The Stage* (Armstrong, 2013), Tom Armstrong criticises the idea of DJs as performers, claiming they *are not* performers and *should not* be expected to be, and even declaring they shouldn't be on stage at all. He states that "we've gotten to a point where two boys are forced to make complete tits of themselves on stage because nobody, from the industry heads to the promoters to the fans, really knows what they're doing there" (Armstrong, 2013) in reference to a live performance by group *Disclosure*, who were pictured with their DJ equipment visibly not plugged in. *Disclosure*, however, are not the sole accused offenders, when it comes to the almost farcical nature of DJs miming live performances, or put more simply: faking it. Amongst the sourced examples include big names in the EDM scene, such as French group *Justice* ("Justice Fake DJ Sets", 2008), as well as David Guetta (Chaîne de FckFakeDJ, 2011).

Armstrong goes on to compare two live performances – one by Rock n' Roll band *Guns N Roses* and one by EDM group *Swedish House Mafia*. *Guns N Roses* were "live, … spontaneous, and the energy from the five musicians resonates with the tens of thousands in attendance…it's brilliant" (Armstrong, 2013). *Swedish House Mafia*, on the other hand were "underwhelming to the point of comedy" (Armstrong, 2013). In both instances, physical gestures were referred to within the dissimilar live styles – *Guns N Roses* use of gestures smiled upon, as drummer "(Steven) Adler bashes the drums like Animal from the Muppets on speed" (Armstrong, 2013) while the gestures employed in *Swedish House Mafia*'s performance do not gather the same sense of praise, as they are described as "three blokes jumping in the air pumping their fists like excited European football fans" (Armstrong, 2013). The stated physical gestures utlised by *Guns N Roses* were directly responsible for producing music, however, while the *Swedish House Mafia* gestures were not, and were purely used as a visual enhancement.

As when it comes to the aforementioned performance of “experimental” electronic laptop-based music, there are those that believe the physicality of the DJ, and legitimacy as performer, within EDM performance, does exist. For example, Richie Hawtin describes what he sees as the physicality of the DJ essentially as interchanging between various types of equipment during the performance (Native Instruments, 2008). In this regard, the interest lies in two areas: the use of alternative equipment being used to perform the music, and the physical movement between the types of equipment within the set-up. Although, once again, it should be noted that the physical gestures used to create and control music are still merely hand-based, while the greater movement of gestures (i.e., switching between equipment) were not manipulating, or performing, the music in any way.

Other forms of visual stimuli have been attempted in performances of EDM, as Ostertag claims; “people who make electronic dance music have been going to great lengths to divert people’s attention from their actual presence: putting on light shows, showing films and videos, and so on” (Ostertag, 2002, p. 12). During his 2011/2012 world tour, celebrated EDM artist *Skrillex* incorporated motion sensing technology into his live performance, which was mapped and used to manipulate animation projected on a big screen behind him, while he, himself, performed the music using standard DJ equipment in a standard manner (“Skrillex live on stage”, nd). In relation to additional visual components, along with a lightshow synced to musical cues, *Deadmau5* goes a step further in his live performance by wearing a large novelty mouse head (“Gizmodo”, nd). This issue of the use of visuals within performance is also discussed by Nick Collins who, when speaking of live laptop performances, claims that “the act of mapping audio to graphics, to live video manipulations… is an area very much open to abuse: visuals are an overpowering medium and can easily detract from a musical performance” (Collins, 2003, p. 76). He does go on to admit that if handled correctly, however, the connection can have great benefits, although, simply by stating the concern in the beginning shows that the method can easily be, and has often been, used unnecessarily and to no great effect.

1.1: Three interactive music systems for the performance of Dubstep music

In order to resolve the issue of Electronic Dance Music performativity, and as part of my Masters by Research project, I will design three distinct interactive music system set-ups, used to perform EDM – and more specifically, a sub-genre known as Dubstep – constructed in ways to focus on the presence of the human body in performance, therefore eliminating the need for light shows and video projection. Each interactive music system, will focus on a different way in which the body, and human gesture/movement, is used to generate sound and control musical parameters. The first system will focus on the physical gestures of instrumentalists; the second system, the physical gestures of a group of choreographed dancers; and the third system, the physical gestures of audience members. In order to develop the proposed interactive music systems, I will be working with the various computer software programs and hardware equipment, which will be discussed in greater detail at the appropriate time. Each system will be demonstrated (either in a performance or workshop setting), documented (visually as well as sonically), analysed, and evaluated. I hope, through the exploration and combination of various compositional and performative techniques, along with the incorporation of new technology, to discover new and exciting ways to perform Dubstep music.

The problem, in regards to EDM performance, may not lie exclusively in the lack of a human body presence. There may be a range of issues that need to be addressed, if in fact there is an initial problem at all – as discussed in the previous chapter. I do not wish to confirm or take sides with any of the aforementioned criticisms, however, as mentioned previously, the incorporation of the physicality of the human body has been proposed as a solution to the suggested shortcomings of EDM performance, and this is the path I, myself, have decided to follow, and explore in detail, for the purpose of my Masters by Research endeavor.

I will use this section to set a narrative of the overall thesis, stating the purpose of forthcoming chapters in preparation of the culminating investigation into each of the three original interactive music systems for the performance of Dubstep music. I will briefly state what will be discussed in each chapter, and how it relates to the setting-up of – and ideology behind – the three systems.

The immediate subsequent subchapter, entitled *Interactive music systems*, cites a selection of interactive music systems – as an overview – developed by various composers in an attempt to counter the criticisms of electronic computer music performance. An early interactive music system and aligned performance will be analysed. This is a necessary preparatory step – in relation to my own undertaking – as I explore the origins of, and numerous ways of applying, various practices I eventually employ in my own systems.

In the following subchapter, *Features of Dubstep music*, I define the genre, describing the fundamental compositional aspects and characteristics. This is important as a reference point, for the reason that throughout the project, and especially in relation to musicality aspects within the interactive music systems, I work within the boundaries of this established definition.

The chapter that follows, entitled *Hierarchy on two levels*, will be divided into two main sections (or subchapters): *Musical phenomena/parameter categorisation* and *Data acquisition/interpretation categorisation*. An overall sound generation and parameter control hierarchy is proposed over the three systems as a whole (i.e., not within each system, but rather over the collective three systems). In order to establish this, I categorise musical phenomena/parameters (examined in the first subchapter) into three groups, as well as ways of interpreting digital information acquired from physical gestural data (examined in the second subchapter) also into three groups. The sorted data within each relevant group, from both methods of categorisation, will be linked together (see Table 1.1). That is, level 1 [*Foreground*] of the 1st method of categorisation linked with level 1 [*Fast moving*] of 2nd method of categorisation; level 2 [*Middleground*] of 1st method of categorisation linked with level 2 [*Medium moving*] of 2nd method of categorisation; level 3 of 1st method of categorisation [*Background*] linked with level 3 of 2nd method of categorisation [*Slow moving*]. (The terms within the square brackets are explained within the chapter). By combining the two categories of information, three 'levels of control' are created, that are subsequently used to determine the main focus and underlying process of control when it comes to developing each of the three systems. Each 'level of control' is applied to one of the three systems each, and therefore operates in conjunction with the physical gestures of instrumentalists, dancers or audience members, depending on the system. This process becomes clearer when discussed in greater detail within the actual chapter. This chapter is essential, as it presents the initial categorising stage of a clear compositional procedure to be followed – conducted in order to determine the fundamental way in which the relevant body, and human gesture/movement, is used to generate sound and control musical parameters within each interactive music system. It can be seen as the template for the gesture-to-sound mapping strategies employed. It can once again also be used as a reference point.

Table 1.1: Types of control applied to each interactive music system

Interactive music system	**Level of control**	**Method of categorisation** (level of sorted information)**:**	
		1. Musical phenomena/parameter	**2. Data acquisition/interpretation**
1	1	Level 1 [Foreground]	Level 1 [Fast moving]
2	2	Level 2 [Middleground]	Level 2 [Medium moving]
3	3	Level 3 [Background]	Level 3 [Slow moving]

The next three chapters, *Evaluation/Analysis of System Performances* (chapters 3, 4, and 5), will be where I document, analyse and evaluate each of the three systems. I will discuss the overall concepts, compositional/development processes, the performances, audience/participant reactions, and possible ways of improvement.

The final chapter, before the conclusion, entitled *Future work: Towards a fully interactive system*, is a theoretical examination of one fully integrated interactive music system, combining the ideology, methods and technology used to create the three original systems. A hierarchy within this one system is also proposed.

In her 2010 paper, *Bridging the gap: Thoughts on computer music and contemporary (popular) electronic music*, Sarah Keith states, "the specific area of contemporary electronic music performance is well suited to researching new interactive possibilities" (Keith, 2010, p. 38). I believe that my Masters by Research project is important because it does just that; I am developing new interactive music systems in order to perform music within the boundaries of a specific genre of contemporary electronic music; Dubstep. New ways of performing this type of music will inevitably lead to new ways of composing, and therefore establish the possibility of expanding the musicality within the genre.

1.2: Interactive music systems

Alternative ways of performing electronic computer music, by moving away from the confinements of the laptop, are constantly being developed. Invariably, this is being achieved by artists' and composers' continued progress in an area, occasionally referred to, as *interactive composing* (Chadabe, 1997) by building Interactive Music Systems. The books *Interactive Music Systems* (Rowe, 1993) and *Composing Interactive Music: Techniques and Ideas Using Max* (Winkler, 1998), along with similar sources such as the conference paper entitled *Understand Interactive Systems* (Drummond, 2009) and *Physical Interfaces in the Electronic Arts: Interaction Theory and Interfacing Techniques for Real-time Performance* (Bongers, 2000) present an extensive overview of the subject of Interactive Music Systems. The concept is explored in greater detail in Bonger's Ph.D. thesis, *Interactivation: Towards an e-cology of People, Our Technological Environment, and the Arts* (Bongers, 2006).

For the purpose of this project, the definition provided by Sergi Jorda provides the most relevancy, in relation to my own three original systems being developed in an attempt to insert the human body into the performance of Dubstep music; In his doctoral thesis, Jorda claims that Interactive Music Systems are computer-based, are interactive, and generate a musical output at performance time, under the control of one or several performers. He also states that Interactive Music Systems must be 'interactive' enough to affect and modify the performer(s) actions, thus provoking an ongoing dialog between the performer(s) and the computer system (Jorda, 2005).

1.2.1: Analysis/overview of Variations V

As the key premise of the idea behind my own original systems is the focus on the different ways human physical gestures can be used to generate and control music, a good starting point would be to look at the revolutionary interactive dance piece, *Variations V* by John Cage. A video of the work being performed can be seen at ("John Cage & Merce Cunningham — Variations V", nd). The work has been thoroughly analysed and extensively written about (Miller, 2001).

Cage worked with dance choreographer Merce Cunningham to develop a system in which there was a relationship between movement and sound, where the dancers "functioned as co-composers" (Miller, 2001), and manipulated the sounds produced, as much as the musicians operating the electronic equipment. In *Cage, Cunningham, and Collaborators: The Odyssey of Variations V*, Leta E. Miller describes how

> The stage set consisted of twelve 5-foot-high antennas Moog had built for the occasion, capacitance devices that sensed the proximity of the seven dancers: …At the base of the antennas were additional sensors, a set of photocells built by Bell Lab engineers under the supervision of the Swedish research scientist Wilhelm (Billy) Kliiver. Whenever the dancers interrupted the light to the photocells or came within a four-foot radius of the antennas, they triggered switching circuitry in the mixer, which in turn fed six loud-speakers spread around the hall. The dancers thus articulated [both] the performance space . . . and the sound space. (Miller, 2001, p. 545-546)

This set-up is a perfect example of an interactive music system, where certain sounds are generated, and allocated parameters are controlled by an input of a physical nature (i.e., gestures/movement produced by human dancers – none of which handle or perform with a tangible musical instrument).

1.3: Features of Dubstep music

Dubstep is a genre of electronic dance music, originating in London, UK. Its sound has been described as "tense, almost oppressively dark" ("All music: Dubstep", 2011) and generally incorporates overwhelming bass lines, sparse rhythm, clipped samples and occasional vocals. The music is frequently written in a minor key, and often features dissonant harmonies created by the tritone interval. The distinctly implied halftime tempo generally sits around the 140bpm mark. A characteristic of many

Dubstep pieces is the use of Wobble Bass, a musical technique created by applying a Low Frequency Oscilator (LFO) to manipulate the filter cut-off frequency of the bass ("All music: Dubstep," 2011; Clark, 2007, 2009; McKinnon, 2007; Sande, 2007). According to Arne Eigenfeldt and Philippe Pasquier in *Towards a generative electronica: Human- informed machine transcription and analysis in Max/MSP*:

> *Dubstep* has a tempo range of 137-142 BPM, with a half-time feel that emphasizes the third beat (rather than two and four). It tends to be rather sparse, with a pre-dominant synthesized bass line that exhibits a great deal of rhythmic low frequency modulation, known as a "wobble bass". (Eigenfeldt & Pasquier, 2011)

I will now list important features/characteristics of Dubstep music as a way to firstly establish the boundaries of the genre to work within, as well as simultaneously tabling key parameters that I control in my originally developed systems. Categorising of the parameters based on scales of time will occur in the following chapters – but for now I will simply list and describe them.

Table 1.2: Dubstep characteristics

<table>
<tr><th>Structural</th><td colspan="2">• Intro, Buildup I, Drop I, Bridge, Buildup II, Drop II, Outro
• Sections of bars that are multiple of four</td></tr>
<tr><th>Textural</th><td colspan="2">• Bass, Synth and Drums
• Combination of layers constantly exploited to signal changes in sections, or to create subtle variation</td></tr>
<tr><th>Temporal</th><td colspan="2">• 136-143 bpm
• 3-7 minutes</td></tr>
<tr><th>Rhythmic</th><td colspan="2">• 4/4
• Wobble Bass
• Sparse, half-time drums – double-time drums during drops</td></tr>
<tr><th>Harmonic</th><td colspan="2">• Minor/pentatonic – dark sounding
• Tritone interval</td></tr>
<tr><th>Timbral</th><td>• Additive, Subtractive, Frequency Modulation types of Synthesis
• ADSR
• Distortion
• Reverb</td><td>• Dirty, grungy synth
• Subbass</td></tr>
<tr><th>Expressional/ Dynamics</th><td colspan="2">• Humanised dynamic variation
• Crescendos/decrescendos</td></tr>
</table>

Structure

The typical structure of a piece of Dubstep music comprises of sections: *Intro*, *Buildup I*, *Drop I*, *Bridge*, *Buildup II*, *Drop II*, *Outro*. This is not necessarily a set-structure, as not every piece will follow it strictly – the order could be different, while certain sections may be excluded. However, when listening to any Dubstep piece, it should be a relatively straightforward exercise to allocate each section one of the above labels (i.e., there may not be an *Outro*, or the *Bridge* may come after the second *Drop*, but the included sections should be able to fit into the listed categories). Usually each section lasts for a number of bars that is a multiple of four (i.e., 8, 12, 16, 32, etc.)

Texture

Dubstep is just another subgenre of the greater Electronic Dance Music, and is therefore a computer-based art form, comprising of digitally created sounds and use – and manipulation – of samples. Therefore, the computer is primarily the sole "instrument", however, distinct instrumental layers are created within it. With this in mind, the three paramount Dubstep layers created digitally are that of *Bass*, *Synth* and *Drums*. Additionally, other layers can be added, such as occasional vocals, or other built-from-scratch-signal-processed or sample-based instrumentation – these are not integral to the characteristic Dubstep sound, however. Like other forms of music, the dropping out and coming in of layers to alter the texture can be used to signify changes of sections, or simply utlised briefly to create subtle variation within the one section.

Temporal

Works of Dubstep primarily lie within the 136-143bpm range. Throughout the piece, however, the tempo remains constant. This is mainly due to the fact that Dubstep is primarily performed in clubs by DJs, who use the technique of mixing to transition from one song into another seamlessly. Therefore, if the work is of 140bpm – or there about – the chance of having the track played in clubs is higher, as the ease of including in a set is greater. The usual duration of a typical Dubstep piece would be in the 3-7 minute range.

Rhythmic

Probably the most unique aspect of all Dubstep music comes under this categorisation; that of which is known as *Wobble Bass*. Wobble Bass occurs when a Low Frequency Oscillator (LFO), of a certain wave shape, is applied to modulate the Synth/Bass layer by rhythmically altering the timbral content. The most common example, for instance, is when an LFO of a sine wave shape controls a Low Pass Filter assigned to the relevant layer. Now even though it is indeed the timbral content (and occasionally additionally frequency and amplitude) that is modified to create Wobble Bass, I have included the feature under *Rhythmic*, as I believe the main focus is on rhythm. Dubstep is a rhythmic and beats-orientated type of music, and the technique is used to drive the composition. Music within the genre is primarily written in a 4/4 time signature – once again, the reason behind this being the ease of transitioning from one track into another in a live mixing set, performed by a DJ, is increased if all tracks are in 4/4. It also creates a sense of predictability for club-goers to dance to. In a rhythmic sense, sparse drums are utilised, which generate a half-time feel. During more intense, and primarily focal points of the track (such as the drop section), however, the drums frequently switch to a double-time pattern, to give the sonic impression that the music is speeding up.

Harmonic

The harmonic content typically is of a minor/pentatonic key, with the use of the tritone interval featuring frequently. The overall sound, from a harmonic aspect, is a dark one.

Timbral

I could go on endlessly listing the timbral characteristics of every layer and every sound, but instead, I will detail the timbral aspects that are constantly being focused on in Dubstep composition, and manipulated in various ways to establish musical variety – once again, this encompasses timbral parameters I also control in my own performances. These ways are tried and tested, feature heavily in music of the genre, and exist within a range of possibilities as to abide by the characteristics of Dubstep musicality.

The focus on timbre is on the built up, or created, initial sound (the obvious example being the dirty, grungy synth sound prevalent in the majority of Dubstep music), as well as how timbre-altering techniques are used to create musical variation and interest. Low subbass is also a necessary layer of orchestration. Different types of synthesis can be applied, including Additive (the laying of sound waves), Subtractive (this is a major one, dealing with the use of all different types of filters), and Frequency Modulation (using one sound wave to alter that of another). The Attack-Decay-Sustain-Release envelopes (ADSR) of each produced sound are paid close attention to, as are effects on each signal, changing the characteristic of the sound, such as distortion or reverb.

Expressional/Dynamics

Contrived natural sounding subtle dynamic variation within melodic phrases, drum tracks, pad sequences, or whatever musical layer it may be, produce a humanised sound within a computer-based genre of music. In order to achieve this effect, the natural sounding variation can be programmed as an algorithm within the Digital Audio Workstation, or alternatively, the composer could physically play a tangible instrument (such as keyboard or drum-pad), sending MIDI information that incorporates the dynamic variation. The latter of the methods, by definition, would be more "humanised", as it is derived from an actual human input. The use of crescendos/decrescendos is also prevalent – usually in a way to develop tension within a *Build-up* section.

1.3.1: Dubstep music

Dubstep music can be found for free on many websites, however, UKF Dubstep, found at ("UKF Dubstep", 2014) provides an extensive collection, and is kept up-to-date by uploading new pieces of music of the genre regularly.

1.3.2: Dubstep performance

As with many matters surrounding live electronic dance music performance; the issue of planning a DJ set is yet another contentious issue. As brushed upon, the favoured method of Dubstep performance is the mixing/transitioning of one song seamlessly into the next (with a variety of performance techniques such as distortion, beat-repeating, reverb, filters, etc. applied at the DJ's discretion). Once again stemming from the idea that the virtuosity and skill required for performances of this nature could come down to the mere push of a 'play' button, the question of how much planning (as opposed to total improvisation) is acceptable is constantly being asked. In his article entitled *Is Planning a DJ Set Cheating?* (Golden, 20120), Ean Golden establishes a spectrum of DJ preparation, starting from lowest preparation to highest:

1. Setting up likely playlists
2. Organising songs by set time
3. Planning an order of songs with variables
4. Pre-configuring every mix down to the second
5. Pre-sequencing a set and playing it back

Assuming most DJ's preparation falls somewhere in the middle, Golden explores various ways to go about the performance practice, calling on his own experiences, as well as and methods and techniques he, himself, employs in live settings to find the perfect balance of preparation and improvisation.

Additionally, and although not in exclusive relation to Dubstep performance, but rather Electronic Dance Music in general, ("ELI5", 2013) provides a knowledgeable article examining typical performance set-ups, including technology used, as well as techniques utilised.

CHAPTER 2

Hierarchy on two levels

This chapter examines the fundamental concepts that make up the interactive music systems developed for the performance of live Dubstep music. To produce an interesting and absorbing interactive music system – for both performer and audience – it is important to have a balance between obvious human-action-to-sound relationships and not so obvious relationships (Winkler, 1998). To do this, I will create a model of gesture-to-sound control of two hierarchies within the systems as a collective. The first hierarchy categorises musical phenomena, and the various encompassed parameters, discussed in a general sense. The second hierarchy categorises the collecting of gestural data electronically, and how this information is interpreted. By applying the two hierarchies simultaneously, each of the three performances has a template – or set of overriding rules – to abide by, which will aid the analysing and evaluating process, used to determine how successful the application of the physicality of the human body was in performing Dubstep in a live environment.

2.1: Hierarchy 1: Musical phenomena/parameter categorisation

In his journal article *Relations between musical and scientific properties of time* (Pressing, 1993), Jeff Pressing organises 'primary musical phenomena' into a time scale. The musical examples – from shortest time span to longest – include single waveform, envelope attack component, single note, steady note production, modulation (vibrato, tremolo), tempo/pulse, motif, phrase, melody, movements, and entire pieces. In addition to this, Curtis Roads, in his book entitled *Microsound*, discusses time scales within music, stating that "a central task of composition has always been the management of the interaction amongst structures on different time scales" (Roads, 2003, p.3). He goes on to distinguish nine separate time scales: (from longest to shortest) *Infinite*, *Supra*, *Macro*, *Meso*, *Sound object*, *Micro*, *Sample*, *Subsample*, and *Infinitesimal*.

The ideas of Pressing and Roads, in relation to musical phenomena in a temporal scale, are used to categorise phenomena – and parameters that make them up, which in turn can be used to alter the relevant phenomena – into a hierarchy of three levels: *Foreground* (FG), *Middleground* (MG) and *Background* (BG). The given titles to each level stem from the notion that multiple phenomena are being presented simultaneously to the listener within any given piece of music; yet it is still possible to distinguish them on a level, based on the *way* they are registered cognitively when heard. That is, *Foreground* phenomena are heard in-the-now, existing on a real-time hearing and registration level; *Middleground* phenomena are realised in hindsight; while *Background* phenomena are realised in greater hindsight, once the piece has concluded (or at least after much of it has been heard), as it is this level that discerns structure. This concept is further alluded to when discussing each level in detail.

Pressing gives an order of musical phenomena, from a shortest to longest time-span, yet he does not categorise into groups (or levels), as I subsequently do for the purpose of my project. Roads, on the other hand, does present groups, or levels in relation to time, of how music phenomena and sound are heard, registered and furthermore recognised by listeners as contributing aspects to the overall piece:

> Musical time can be viewed in terms of a hierarchy of time scales… Mathematical trees are inverted, that is, the uppermost level is the root symbol, representing the entire work. The root branches into a layer of macrostructure encapsulating the major parts of the piece. This second level is the form: the arrangement of the major sections of the piece. Below the level of form is a syntactic hierarchy of branches representing mesostructures that expand into the terminal level of sound objects. (Roads, 2003, p.11-12)

I have not attempted to further the research by Pressing or Roads, but instead, merely adopt the concepts to establish hierarchies – or templates – in which to create environments to explore new and exciting ways to perform Dubstep music by inserting the physicality of the human body.

Table 2.1: Musical phenomena/parameter categorisation

Level	Roads categorisation	Pressing musical parameter	Pressing Time scale
1. Foreground	• Sound object • Micro	Single Waveform	0.00005 – 0.05 s
		Envelope Attack Component	0.0005 – 10 s
		Single Note	0.001 – 10 s
		Steady Note Production	0.05 – 10 s
		Vibrato/Tremelo	0.1 – 0.5 s
2. Middleground	• Meso	Tempo/Pulse	0.1 – 5 s
		Motif	0.5 – 5 s
		Phrase	3 – 30 s
		Melody	5 – 100 s
3. Background	• Macro	Movement	30 – 1000 s
		Piece	30 – 30000 s

Table 5.1 presents the three levels (*Foreground*, *Middleground* and *Background*) established for this hierarchy, while categorising musical phenomena and parameters (and their respective scales of time) discussed by Roads and Pressing into each one. The phenomena/parameters that I categorise into FG, MG and BG levels are ultimately eligible for live manipulation, controlled by physical human gestures during each performance. This hierarchy is applied to the Performances of live Dubstep as a whole – so that the methodology of *Performance 1* incorporates *Level 1* of the hierarchy (FG); *Performance 2* incorporates *Level 2* of the hierarchy (MG); while *Performance 3* incorporates *Level 3* of the hierarchy (BG). That is, the allocated level presents the only level of phenomena/parameters eligible for control for each respective performance, while the two alternative groups will be constant and uncontrollable in a performance matter, as detailed in the table below:

Table 2.2: Parameter eligibility

Performance	Parameter level eligible for control	Parameter levels ineligible for control
1	Foreground	Middleground, Background
2	Middleground	Foreground, Background
3	Background	Foreground, Middleground

It should be noted that, even though I have intended to establish three distinct levels (FG, MG and BG) containing musical phenomena/parameters that fit within the defining boundaries, unfortunately, the task is not so black-and-white. In relation to his own musical time scale groupings, Roads states “in some cases the borders between time scales are demarcated clearly; ambiguous zones surround others” (Roads, 2003, p.4). He has come to find that

> …compositions may contain overlapping elements (on various levels) that cannot be easily segmented. The musical hierarchy is often fractured. Indeed, this is an essential ingredient of its fascination. (Roads, 2003, p.12)

Roads provides an example of such uncertainty, by comparing the musical concepts of pitch and rhythm, and finding a grey crossover area:

> The low-frequency boundary for the sensation of a continuous sound – as opposed to a fluttering succession of brief microsounds – has been estimated at anywhere from 8 Hz (Savart) to about 30 Hz…Between the sensation of a continuous tone and the sensation of metered rhythm stands a zone of ambiguity, an infrasonic frequency domain that is too slow to form a continuous tone but too fast for rhythmic definition. (Roads, 2003, p.17)

Unsurprisingly, these grey areas arose during my own attempt to categorise musical phenomena/parameters into clear-cut and distinct levels, especially when designating specific phenomena/parameters characteristic to Dubstep music. When these cases do occur, however, a definite position is taken on the inexact phenomena/parameter, and a justification for the decision is given.

2.1.1: Foreground category

The first level of categorised musical phenomena in a time scale is known as the *Foreground* level. This level contains musical phenomena that is seemingly heard in real-time as it happens – or heard “in the now” – and is on the shorter end of the time-span scale. The phenomena is played out in such a short amount of time, the listener registers and interprets the sonic information simultaneously (or close enough) to it being produced – that is, past or future sonic events do not contribute to the understanding and perception of the phenomena. Roads examines this concept of the physical NOW, while discussing Fritz Winckel’s “thickness of the present” notion

(Winckel, 1967; Roads, 2003, p.4). He mentions how "this temporal interval (0-600ms) constitutes an estimate of the accumulated lag time of the perceptual and cognitive mechanisms associated with hearing" (Roads, 2003, p.4).

In order to establish the boundaries of the *Foreground* category, I interrelated the ideas put forth by both Pressing and Roads. Taking the phenomena listed by Pressing; within this category, I have included single waveform, envelope attack component, single note, steady note production, modulation (vibrato, tremolo) (see Table 5.1). When dividing Roads' groupings, I have included *Sound object* and *Micro* within this hierarchical category[1].

Roads' *Sound Object* category "encompasses events of a duration associated with the elementary unit of composition in scores: the note" (Roads, 2003, p.16), which he states "usually lasts from about 100 ms to several seconds" (Roads, 2003, p.16). Traditional notes played by conventional instruments are not the only type of sound object, however, as "the concept of sound object extends this to allow any sound, from any source" (Roads, 2003, p.17). Roads allocates four properties to a sound object, or traditional singular note[2]. The four properties are *Pitch*, *Dynamics*, *Timbre* and *Duration*, with the last property existing "generally between ~100 ms (slightly less than a thirty-second note at a tempo of 60 M.M.) to ~8 seconds (for two tied whole notes)" (Roads, 2003, p.18). As discussed, all properties, in the context of being encompassed in a sound object or singular note, are registered simultaneously and in-the-now when heard. In regards to timbre, Roads notes, "the sensation of tone – sustained or continuous event of definite or indefinite pitch – occurs on the sound object time scale" (Roads, 2003, p.17). Additionally, the Attack-Decay-Sustain-Release progression over the duration of a sound object (applied in relation to amplitude, frequency, a variation of a timbral quality, or special aspect) can further

[1] The *Sample*, *Subsample*, and *Infinitesimal* groups define musical experiences of even shorter time-scales, so by default they also correlate with this *Foreground* category, as they deal with phenomena that is heard and interpreted in real-time. However, I have limited the discussion to include only *Sound object* and *Micro*, as they relate more closely to the boundaries of my own category, and the manipulating of these parameters of the shortest duration will not be applied in my own interactive music system.

[2] Roads makes a distinction between traditional notes – with their various time-honored restrictions – and sound objects in a greater sense. Sound objects' properties can differ between seemingly exact notes, while those of traditional notes cannot. For the purpose of my project, however, for now this observation of distinction is irrelevant, as the understanding of given properties is essentially all that is required.

breakdown the parameters of a singular note. Roads mentions this, as well as how changes within single sound objects can create musical variation:

> The properties of a sound object are free to vary over time. This opens up the possibility of complex sounds that can mutate from one state to another within a single musical event. In the case of synthesized sounds, an object may be controlled by multiple time-varying envelopes for pitch, amplitude, spatial position, and multiple determinants of timbre. (Roads, 2003, p.19)

Roads' *Micro* time scale group consists of "transient audio phenomena, a broad class of sounds that extends from the threshold of timbre perception (several hundred microseconds) up to the duration of short sound objects (~100 ms)" (Roads, 2003, p.20-21). He discusses the idea of interpreting sound, and its various parameters, in real-time:

> Human beings need time to process audio signals. Our hearing mechanisms impose minimum time thresholds in order to establish a firm sense of the identity and properties of a microevent… If one impulse follows less than 200 ms after another, the onset of the first impulse will tend to mask the second, a time-lag phenomenon known as forward masking, which contributes to the illusion that we call a continuous tone. (Roads, 2003, p.22-23)

After establishing a category of *Foreground* phenomena/parameters on the shorter end of the musical time-scale, and examining various general examples and how they fit within this level, it is time to specify how they are applied in a hierarchical manner within the originally developed interactive music systems for the performance of live dubstep music.

Performance 1 presents a live performance of Dubstep music, where physical gestures are utilised in numerous ways, and in conjunction with various forms of technologies, to control the *Foreground* phenomena/parameters of the music. In saying this – and after considering the corresponding ideas put forth by Roads and Pressing in relation to musical phenomena on the shorter end of the time-scale – the musical aspects that will be eligible for control, under the category of *Foreground*, will be *Frequency*, *Amplitude*, *Modulation* on the shorter end of the time spectrum (such as tremelo and vibrato), and the infinite parameters that correlate with and affect *Timbre*. These manipulated properties of sound will be encompassed within short time frames that are processed by the listener in real-time, as in singular sound objects or notes, and within their ADSR's.

2.1.2: Middleground category

The next level of categorised musical phenomena in a time scale is known as the *Middleground* level. This level contains musical phenomena that sit within the centre regions of the time-span scale. The phenomena exist as groups of elements –say, blocks – comprehended only when taking into account the numerous encompassed events or components making up the blocks, and are realised as patterns or structures. That is to say, past and future musical events contribute to the understanding of these *Middleground* phenomena. So, unlike *Foreground* phenomena, which are registered, and interpreted simultaneously to the sonic information being heard, *Middleground* phenomena are made sense of in hindsight and over a longer period of time. Speaking in relation to the aforementioned sound objects, Roads explains how on this middleground type of level, sound objects are grouped into somewhat of a hierarchy of phrase structures of durations measured in seconds, and that "it is most often on (this) meso level that the sequences, combinations, and transmutations that constitute musical ideas unfold" (Roads, 2003, p.14).

Taking the phenomena listed by Pressing; within this category, I have included single tempo/pulse, motif, phrase, and melody (see Table 5.1). When dividing Roads' groupings, I have included *Meso*.

When discussing his *Meso* level, Roads examines the concepts of melody and harmony, as well as rhythm in the given context of how they are realised by the listener:

> Melodic, harmonic, and contrapuntal relations happen here, as do processes such as theme and variations, and many types of development, progression, and juxtaposition. Local rhythmic and metric patterns, too, unfold on this stratum. (Roads, 2003, p.14)

Once again I will now detail how the musical phenomena/parameters within this *Middleground* category are utilised and treated in a hierarchy within the interactive music system. *Performance 2* presents a live performance, where physical gestures are used to control the *Middleground* phenomena/parameters of the music. The musical phenomena, which lie in the middle regions of the time-scale, that are focused on, surround motif/melody/phrases, as noted by Pressing. The musical aspects that are eligible for control, under the category of *Middleground*, however, are aspects integral to the construction of motif/melody/phrases, being *Rhythm* and *Scale*. To be more specific, the method of control centres on the choice and possible options for various permutations of rhythm and scaling (applied in melodic, as well as harmonic ways). This concept becomes more apparent when the performance and system are thoroughly discussed and analysed. The manipulation of these musical aspects are executed in ways so that musical variation is realised by the listener in hindsight, as opposed to "in-the-now", by controlling how various individual elements of the motif/melody/phrase relate to each other in a rhythmic or scale-wise sense.

2.1.3: Background category

The final level of categorised musical phenomena in a time scale is known as the *Background* level. This level contains musical phenomena on the longest side of the time-span scale. Like the phenomena of the previously discussed *Middleground* group, *Background* level musical phenomena are once again perceived in hindsight, however, in this case, over a much longer period of time. Whereas *Middleground* phenomena are realised by recognising the relationships between short sound-object-length musical elements (found within *Forground* phenomena), *Background* phenomena are understood by identifying comparisons between all musical aspects, including ones of far greater length and weight. Roads claims that this "level of musical time corresponds to the notion of form, and encompasses the overall architecture of a composition" (Roads, 2003, p.141).

Taking the phenomena listed by Pressing; within this category, I have included movement and piece (see Table 5.1). When dividing Roads' groupings, I have included *Macro*. Technically, the remaining groups of *Supra* and *Infinite*, established by Roads, which detail progressively longer musical phenomena, would also fit in this category. However, as the manipulating of these parameters of the longest duration is not applied in my own interactive music system, it serves no purpose to examine them in any detail.

When examining the *Macro* time-scale, Roads talks about how listeners understand this level of the musical hierarchy:

> Unless the musical form is described in advance of performance (through program notes, for example), listeners perceive the macro time scale in retrospect, through recollection. (Roads, 2003, p.11)

In regards to how I approach the establishing of a hierarchy within my own interactive music systems, *Performance 3* presents a live performance, where musical phenomena of great length and weight of the *Background* level, along with their parameters, are controlled by physical human gestures. The musical aspects eligible for control, under the category of *Background*, are *Texture* and *Structure*. That is to say, the parameters responsible for the execution of *Texture* (being what instrumental layers are present and when) and *Structure* (being the length of the various sections) within the music produced are determined by human gestures during performance.

2.2: Hierarchy 2: Data acquisition/interpretation categorisation

As technology has evolved, the possibilities of digitally collecting gestural information have increased, while the accessibility has become even greater. Much research has been undertaken in this area, studying a range of methods by utilising equipment such as microphones[3], cameras, the Xbox Kinect ("Xbox 360 + Kinect", 2013), Nintendo Wiimotes ("Wii", 2013), and various MIDI-sensor technology to track and capture motion and movement, human or otherwise (Overholt, Thompson, & Putnam, 2009; Mulder, 1994; Ng, 2004; Winkler, 1995; Kiefer, Collins &

[3] Even though microphones cannot be used to track visual information of a physical gesture, they can, however, capture the direct sonic output that a gesture may produce. For example, in regards to singing, the observable physical movement of the mouth (as well as the unseen but necessary movement of specific organs) cannot be captured by a microphone – however, the direct sonic result of these physical gestures (i.e, a sung note) can be captured.

Fitzpatrick, 2008; Odowichuk, Trail, Driessen, Nie, Page, 2011; Jessop, 2010; Vickery, 2002). In this context, my intention was not to necessarily further the research done in these areas, but in fact, it was to use many of the same methods and techniques examined as a foundation for my project.

There is a second hierarchy that is applied to the overriding methodology of the originally developed interactive music systems for the performance of live Dubstep music. This hierarchy concerns how physical human gestures are interpreted during performance and subsequently utilised to generate/control music – that is, while the gesture/s are being tracked, what digital data is acquired, and furthermore what aspects of the obtained information are used to control allocated parameters of music, and how this is achieved in each case, and by extension, in each of the three performances. The hierarchy will be made up once again of three categories: *Fast Moving Information*, *Medium Moving Information*, and *Slow Moving Information*.

This hierarchy will once again be applied to the Performances of live Dubstep as a whole, in conjunction with the originally proposed hierarchy – so that the methodology of *Performance 1* will incorporate *Fast Moving Information*; *Performance 2* will incorporate *Medium Moving Information*; while *Performance 3* will incorporate *Slow Moving Information*. That is, the allocated level presents the only level of data acquisition/interpretation applied for control for each respective performance, while the two alternative levels will not be applied or utilised for control of the music, as detailed in the table below:

Table 2.3: Data acquisition/interpretation application

Performance	Data acquisition/ interpretation level applied	Data acquisition/ interpretation level not applied
1	Fast moving	Medium moving, Slow moving
2	Medium moving	Fast moving, Slow moving
3	Slow moving	Fast moving, Medium moving

The size/importance/obviousness of gestures is also an aspect of the hierarchy, however, once again, only when considering all the performances as a collective. That is, gesture size/importance/obviousness are not part of a hierarchy within each separate performance.

The use of the term ‘gesture’ is a broad term in live electronic music performance of all sub-genres – as has been explored by Alexander Jensenius in his paper *To Gesture of Not? An Analysis of Terminology in NIME Proceedings 2001-2013* (Jensenius, 2014). ‘Gesture’ is defined within this study as movement of the human body, but specifically – and in fact interchangeably – in regards to what is being tracked and used for musical control. The difficulty in confining the term is due to the way that gestures are treated differently in each performance. For example, the MacMillan dictionary’s (Gritten & King, 2006) definition of ‘gesture’ (as noted in Jensenius, 2014) is broadened to include “a movement that communicates a feeling or instruction”. As is discussed in greater detail in subsequent chapters, this definition would indeed apply to the gestural approach undertaken in *Performance 2*, as the human movements tracked for musical control are of an expressive and more full-bodied nature. In saying this, the same definition would not, however, apply to the approach in *Performance 1*, as the physical gestures tracked for musical control can be on such a small scale that the human performer may execute them completely inadvertently. Finally, the MacMillan definition also struggles to relate fully to the approach in *Performance 3*, as the gestures tracked for musical control are not of limb movements of the one human participant (no matter how expressive), but rather the movements and positions in space of multiple participants as a collective whole. Throughout the current *Hierarchy 2: Data acquisition/interpretation categorisation* subchapter, gestures are referred to in a general sense. However, each performance case – in regards to the gestural approach undertaken (including what constitutes the period of a gesture within the context of each performance) – is examined more extensively in the relevant performance evaluation/analysis chapter (chapters 3 – 5). Just to reiterate, the term ‘gesture’ – within the scope of this project attempting to cover all bases – is not easily definable.

Within each performance, various forms of technology (such as microphones, cameras, etc.) are set up, as part of the relevant interactive music system, to focus on and track allocated physical gestures. This information is then processed within computer software – primarily MaxMSP – and converted into digital data, being a set of representative number values. That is, as the various gestures move and change, the representative number values follow suit. The point of the gestural-acquisition hierarchy is to determine how these sets of numbers, within each performance, are interpreted and further used to generate sound and control parameters of music within the live environment. More precise set-ups of each performance (including data extraction, scaling and mapping) are detailed in the relevant performance evaluation/analysis chapters (chapters 3 – 5), while the purpose of this chapter is merely to introduce and outline the main points of this second hierarchy, which acts as one component of the overall methodology undertaken in order to accomplish live performance.

In the case of all points, there exist objects within MaxMSP that can be used for the appropriate observation and extraction of data from gestures, in conjunction with the original tracking device (be it microphone, camera, etc.). However, the originally received data, expressed as ever-changing numerical information – will be of a raw type. For example, volume is expressed in decibels, pitch is expressed in frequency, and timbral information is expressed as the raw data of whichever aspect that was chosen to be extracted. It is best now, to think of this collected information now as only number values, as when the time comes for them to be used as a device for further control, they will be used to automate completely different parameters of music than the ones the data was originally acquired from. In order for this numerical information to be used for further MIDI automation of other layers' parameters, each raw data output is divided accordingly, and scaled to fit within a 0–127 range. The collected gestural data, now converted to numeric values in the range of 0–127, is now ready for further MIDI control of other layers' parameters (i.e., ready to be mapped). In this way, human movement is able control musical parameters and perform live Dubstep music.

2.2.1: Fast moving information

The first category of the three-tier hierarchy is known as *Fast Moving Information*, and will apply to the overriding procedure of the interactive music system utilised in *Performance 1*. This level, as the name suggests, processes incoming digital data, and applies it to create and control music in a fast manner. The notion of specific and detailed gesture tracking, focusing on the minutia, are also explored, where subtle variation in movement affects control and sound manipulation.

As mentioned in previous chapters, *Performance 1* focuses on the physical gestures of instrumentalists, and the musical parameters eligible for control are those of the *Foreground* variety. The human gestures utilised are on the smaller to medium scale of movement. Each instrumentalist is tracked individually by respective technological equipment, and certain observed movements and gestures they employ are captured, and then inputted into MaxMSP in the form of digital data.

This digital information, processed and converted into a continuous stream of numerical values, ready to be used for further control of musical parameters, is interpreted in a fast-moving sense in the way that gesture-to-sound control occurs in real-time. This involves the continuous extraction of data during the entire period of the observed gesture, to be used for simultaneous continuous automation of an assigned musical parameter.

As mentioned previously, certain gestural data is extracted to produce a stream of numerical information that is converted into a range of 0–127 in order to abide by the overall MIDI-based approach of the interactive music system. With this in mind, the ratio between the output of the physical gesture number values and the input of the manipulated *Foreground* parameter of music does not need to be a 1:1 relationship – That is, for example, as the output value moves from a range of 15–40 over a period of five seconds, the input value acts in the exact same way. As we eventually come to see, the relationship is sometime reversed, so that as the output value moves from a range of 15–40 over a period of five seconds, the input value moves from 40–15 over the same amount of time. Furthermore, the relationship is occasionally applied at a constant ratio rate (i.e., a multiplier) – say, 1:3 – so that as the output value moves from a range of 15–40 over a period of five seconds, the input value moves from 45–120 over the same amount of time[4].

[4] The techniques described here are applied in more complex ways in Performance 1, with the specifics (including mapping) detailed in the evaluation/analysis chapter. However, for the purpose of providing a clear understanding; basic numerical values and easy-to-follow processes are given as theoretical examples of the techniques here.

In each of the above gesture-to-sound approaches, in regards to ratio set-up, the mapping strategy is that of one-to-one[5]. That is to say, the movement of a physical gesture may be set-up to control one singular parameter.

2.2.2: Medium moving information

The next category is known as *Medium Moving Information*, and applies to the interactive music system used in *Performance 2*. This level processes incoming digital data, and applies it to manipulate parameters of music in a medium-paced manner. In the previous *Fast Moving* category, the smallest of subtle gestures – even incidental movements – affect the extracted information, and subsequently influence not only the converted numerical data, but also the tangible sonic output of the musical parameter. Within the category of *Medium Moving Information*, however, subtleties of this nature would not have the same effect.

Performance 2 centers on physical gestures portrayed by dancers, while the *Middleground* parameters of the music are eligible for control. The human gestures utilised are on the medium to large scale of movement. That is, the movements are of a more full-bodied, exaggerated, and obvious manner than what occurs in *Performance 1*. Once again, each individual participant – this time being dancers rather than instrumentalists – is tracked using a specific form of technology, with extracted information from certain gestures being sent for further processing into MaxMSP.

This digital information, represented now as numerical values, is interpreted in a medium-moving way in the sense that the allocated musical parameter would not necessarily be changing simultaneously, and in a real-time sense, with the respective gesture movement, as would be the case within the *Performance 1* set-up. Continuous extraction of data occurs during the entire period of the observed gesture, however, the methods in which the converted numerical streaming data is used for control is not of a simultaneous automation nature. That is, certain requirements must be met within

[5] Andy Hunt, Marcelo M. Wanderley and Matthew Paradis discuss the importance of mapping in their paper *The importance of parameter mapping in electronic instrument design* (Hunt et al. 2002), and explore the different possible mapping approaches. They discover that one-to-one strategies – if not handled correctly and creatively enough – can have the potential to be too simplistic, and therefore bore the player of the instrument.

the gestural action in order for sonic changes to occur; simply moving – depending on speed and distance – will not necessarily produce any manipulation to the allocated musical parameter.

A MIDI-based approach is once again undertaken for this performance. So, using MIDI value restrictions, three cases of how medium-moving interpretation apply are as follows[6]:

Certain data acquired from a physical gesture produce MIDI values increasing from 0-127 as the gesture is fully executed, however, manipulations to the allocated *Middleground* parameter only occur once the values enter a certain range. For example, depending on how the specifics are set-up, throughout the execution of the full gesture, sonic changes are only triggered during certain stages of the action – there could be four stages, divided equally, so that while the gesture progresses from the first stage into the second, representative MIDI values move from the range of 0-31 (corresponding to the first stage) into the range of 32-63 (corresponding to the second stage), thus activating the trigger, and subsequently sonically affecting the allocated musical parameter. Once the physical gesture continues into the third and fourth stages (represented by the MIDI ranges 64-85 and 86-127 respectively) further musical parameter control occurs. It should be noted that the parameter manipulation is only triggered once the MIDI value moves between the set ranges. This means that while the gesture is in action, even if the dancer is physically moving, until that crossover occurs, no audible change to the parameter will be heard. It is this idea that makes this data acquisition and control category one of a medium-moving level.

Another application of this level used to control musical parameters in a live environment is the method of the extracted numerical data obtaining a certain average value in order to initiate sound manipulation. This concept makes it possible for the speed of certain physical gestures to control musical parameters. For example, MIDI values representing specifically extracted data from a gesture must cover a range of

---

[6] The techniques described here are applied in more complex ways in Performance 2, with the specifics (including mapping) detailed in the evaluation/analysis chapter. However, for the purpose of providing a clear understanding; basic numerical values and easy-to-follow processes are given as theoretical examples of the techniques here.

10-20 in a time period of 0.3 seconds or lower in order to trigger control of an allocated parameter. Needless to say, the dancer could perform the identical gesture, covering the MIDI value range of 10-20, but if this is not executed under the specified time period of 0.3 seconds, no triggered live manipulation of the music occurs.

The final way this medium-moving level of data extraction and control is applied is in regards to multiple and separate gestures working in tandem. This enables a dancer to perform a series of gestures, at any singular speed, where it is the time *between* the executed gestures – as opposed to the speed of the action of the sole gesture –that determines sound control. For example, the system set-up determines that executions of a number of certain gestures trigger the same MIDI value – for all intents and purposes, acting as a singular activate signal. A time period is also allocated between registration of the gestures, so that if the registered time between the gestures is less that 1 second, one form of sound manipulation is triggered; if the time between gestures is between 1-3 seconds, another form of sound manipulation is triggered; and if the time between gestures is greater than 3 seconds, a different form of sound manipulation is triggered. This enables the singular action of the gesture to be executed at any speed while not altering any musical parameters, as well as enabling the dancer to perform alternating permutations of the gestures in order to receive the same audible result, as long as the time between gestures are matching. That is, performing *Gesture 1* four times in a row, with two seconds between actions achieve the same result as performing *Gesture 1* then *Gesture 2* then *Gesture 1* then *Gesture 2* also with two seconds between actions.

Of the three possible applications given, the underlying gesture-to-sound mapping strategy differs between them. The first two cases would be regarded as having a one-to-one mapping strategy, while the final case would have a many-to-one strategy – that is, it takes multiple gestures to control the one parameter of music[7].

[7] In saying this, the use of the term here is a little unorthodox. Typically, a many-to-one strategy entails the ‘many’ to be individually mapped to the ‘one’, usually interacting in complicated and interesting ways (see Drummond, 2009; Hunt & Kirk, 2000; Miranda & Wanderley, 2006). In my case, the ‘many’ *must* be carried out – and performed in sequence – to control the ‘one’. Alternatively, the sequence of gestures itself could be seen as a ‘singular’ recursive gesture controlling the one parameter of music – and therefore the relationship would instead be that of a one-to-one. This ambiguity and interchangeable definition depending on point of view once again highlights the difficulty in categorically delineating what constitutes a gesture within this extensive project.

2.2.3: Slow moving information

The final category, entitled *Slow Moving Information*, applies to the overriding system of *Performance 3*. This level extracts gestural information, and applies it for musical control in a slow manner. In the two previously explained categories, the concept of individual's gestures being utilised for control is applied, whereas, in this category, the idea of gestures, movement, and position in space of the collective of individuals is investigated. That is, singular actions from an individual have no impact on musical parameter manipulation.

Performance 3 focuses on the gestures of human bodies in the audience, with the *Background* parameters of music being controlled in a live way. The gestures utilised are on the larger scale of movement. The audience as a whole are tracked, observing their physical gestures as a collective, by particular forms of technology. The position of multiple bodies in space is monitored, with selected information being acquired and sent for further processing into MaxMSP.

This data, now represented as a continuous stream of numerical values, is interpreted in a slow-moving way in the sense that even though the extraction of gestural data is constant, it is only utilised for musical control during single points – or snapshots – in time. That is, the idea of a 'window of opportunity' method of parameter control is put into practice. The singular points in time, utlised for control, are taken at regular, however, sparse intervals. The controlled *Background* parameters also do not respond simultaneously to when the snapshot of values is taken – instead, the values are registered and called upon for control a period of time later. This is in stark contrast to the method of control applied in the first category, and corresponding performance, where gestures of a singular individual are constantly tracked, and any movement is utilised for control simultaneously.

MIDI is once again employed: so detailing a case of how this slow-moving method of control is applied, using MIDI values would be as follows[8]:

[8] The techniques described here are applied in more complex ways in Performance 3, with the specifics (including mapping) detailed in the evaluation/analysis chapter. However, for the purpose of providing a clear understanding; basic numerical values and easy-to-follow processes are given as theoretical examples of the techniques here.

Specified data extracted from the physical movement and position in space of the entire audience members as a collective whole produces a singular continuous stream of MIDI values. At the start of every 16 bars, for example, a snapshot of the stream is taken, deriving a singular MIDI value each time. This value is then be used to manipulate an allocated *Background* parameter of music at a future point in the live performance. Needless to say, regardless of the movement generated by the audience members, or their positions in space, unless it is at the point in time when the snapshot of values is taken, their gestures and actions do not contribute to the direction of piece, or more specifically the manipulation of *Background* musical parameters. This concept makes this particular level the hardest to actively control and have a compositional say in the music produced.

The above application of this level of data acquisition and musical control has a many-to-one underlying gesture-to-sound mapping strategy. This is due to the fact that a range of multiple gestures taken from multiple individuals are then consolidated into – and interpreted as – one value, used to control one *Background* parameter of music.

2.2.4: "Specified data"

While discussing the three categories of the three-tiered data acquisition and control hierarchy – *Fast*, *Medium*, and *Slow Moving Information* – various possible examples of how each level could be applied were given. While detailing these examples using MIDI values, the phrase "certain data extracted from the gestures…" or some variation, was used in each case. This is because the precise gesture, or what aspect of it was observed for data extraction, is not important at this stage – instead, solely *how* this data was put to use for musical control. The specifics are examined in coming chapters on each performance evaluation[9].

[9] In saying this (and to provide basic examples to use for reference), possible aspects of gestures that are focused on for data acquisition could be the X, Y or Z position of a limb, or the frequency or amplitude information derived as a tangible result of a physical gesture, such as the singing of a note, or the pluck of a guitar string.

CHAPTER 3

Evaluation/analysis of system performances: System 1: Instrumentalists

The first performance in the three part series, which attempts to incorporate the physicality of the human body into live Dubstep music performance, focuses on the gestures utilised by instrumentalists. Within this performance and system, more subtle gestures are exploited to control musical parameters that are registered by listeners as “in-the-now”. The interactive music system is set up so that the performing instrumentalists focus solely on playing their own instrument, however, processes are in place so that gestures they employ affect allocated musical parameters of alternative layers of music. Joel Chadabe claimed “the challenge for computer music composers in the near future will be to use their elite knowledge and skill to create situations in which members of the public without that knowledge and skill can participate meaningfully in a musical process” (Chadabe, 2000, p. 11). This directly relates to the construction of this first system in the way that, in various cases, traditional instrumentalists perform their instruments in traditional ways, while having no real understanding of the computer algorithms taking place in the manipulation from the conventional gestural output of their instruments to the contemporary electronic music result. Nor does the process even *need* to be fully understood for the players to “participate meaningfully”. The use of human bodies participating in gestures associated with music making (e.g., the strumming of a guitar, or the singing of a note) inserts a human presence into the performance, while the music produced is that of Dubstep music.

Three human instrumentalists, playing three distinct instruments, perform a different layer of orchestration each. The instruments utilised represent three diverse types:

1) <u>Vocals</u> – Traditional
2) <u>ReacTable</u> – Electronic
3) <u>Electric guitar</u> – Hybrid

The *Traditional* type contains a conventional and fundamentally non-electronic instrument, the voice; the *Electronic* group contains a more recently developed interface controller, connected to a laptop, the reacTable (see Appendix 1); while the *Hybrid* group contains a traditional-instrument-like interface, used only as controller connected to a laptop, (which will be explained in more detail shortly), the electric guitar. Each instrument is performed by a single human player. A range of primarily different instruments was favourable as it provided a greater scope of exploration for gestures in performance.

As each instrumentalist performs – and contributes to – their own orchestration layer, but also to the live manipulation of the alternative two layers, a human quality into the performance of Dubstep music is therefore embedded. This applies most obviously in relation to the *Traditional* and *Hybrid* performers, as, due to the algorithmic parameter control system that is in place, the instrumentalists would be able to deliberately control many obvious aspects of their own layer of orchestration, while other minor aspects, although technically determined by the vocalists input into the microphone, or the gestures associated with playing guitar, would be harder to control. That is, for example, a performer of these types of instruments can knowingly control a roundabout rhythm of a melody, however, it would not be humanly possible to produce an exact and perfectly times rhythm for a certain amount of time – thanks solely to the nature of being human.

3.1: System set-up/analysis

The main objective of *Interactive Music System 1* is to insert the physicality of the human body into live Dubstep performance, by enabling instrumentalists of various forms to control parameters of music in non-conventional ways. It was a conscious decision to exploit traditional instruments, as the aim of the entire project is to insert a human presence into Dubstep performance. Live bands, as well as traditional instrumentalists rarely, if ever, draw criticism for lack of a physical visual component to their performance.

When it comes to analysing a system and performance of this nature, it is difficult to study one aspect at a time with no crossover, as occasionally an overall knowledge of the system and underlying process as a whole needs to be understood in order to fully examine one component. So therefore, speaking in a general sense, three instrumentalists play together as a band in a live setting. They interact amongst themselves, as live performing groups do. They are focused solely on playing their own instrument, and making their own produced layer of orchestration sound as good as possible. Various forms of frequency, amplitude and visual-symbol tracking technology and techniques are utilised in certain ways in order to control *Frequency*, *Amplitude*, *Timbre* and *Tremelo/Vibrato* of all orchestration layers of a typical Dubstep piece. The system, and aligned performance, is interactive in the sense that the instrumentalists perform in a live "jamming" environment, affecting parameters of music of all orchestration layers, which in turn they would then react to with their own playing and the decisions they make in an instrumental performance sense.

3.2: Application of hierarchies (Level of control 1)

The interactive music system utilised in *Performance 1* applies the *Foreground* level of the *Musical phenomena/parameter* hierarchy in conjunction with the *Fast moving* level of the *Data acquisition/interpretation* hierarchy (See Table 3.1 detailing the system's overriding level of control).

Table 3.1: Performance 1 level of control

1. Musical phenomena/parameter		2. Data acquisition/interpretation	
Level eligible for control	**Levels not eligible for control**	**Level applied for control**	**Levels not applied for control**
Foreground	Middleground, Background	Fast moving	Medium moving, Slow moving

3.2.1: Hierarchy 1: Foreground phenomena/parameters

In regards to the first hierarchy – which dictates the type of musical phenomena and parameters that are eligible for control – the gestures utilised in *Performance 1* control phenomena/parameters of a *Foreground* level. Here, I will not discuss how the parameters are controlled, but merely *what* parameters are controllable, as well as compositional decisions, including established boundaries, which were made in a musicality sense. This hierarchy relates to what is heard as part of the performance in relation to the respective level of phenomena/parameter categorisation.

In order to set up a system in which *Frequency*, *Amplitude*, *Timbre* and *Tremelo/Vibrato* aspects of the generated music are controlled in unconventional ways by use of physical gestures, certain boundaries of control were required to be established. These boundaries were to ensure the music retained a characteristic Dubstep sound, and included the idea that set-up of the system works at its best when the *Middleground* and *Background* phenomena are of a fundamentally Dubstep nature. Because the system exists when human performers play instruments, the musical choices are ultimately down to them, as opposed to the other two performances (*Performance 2* and *Performance 3*), where the generation of music (and ability to stop and start the music) is largely inherent in the computer components of each system. This, therefore, made for a difficult task within *Performance 1* to retain a Dubstep feel – and, in a way, the score and instructions given to players is a necessary element of the system to achieve the Dubstep sound.

The process of establishing genre-defining boundaries was a complex one, due to the nature of the system, and mainly the fact that human players (with minds of their own) are responsible for the production of the music. It should therefore be noted that it is possible for this system to produce music other than Dubstep, depending on the instructions given to the players. In saying this, in this documented case, a score was supplied, instructing players to play in a certain style (of that detailed in *Features of Dubstep music* subchapter), regarding the *Middleground* and *Background* phenomena of fundamentally *Rhythm*, *Scale*, *Structure* and *Layers*. Because these phenomena/parameters were pre-determined and of an instructed style – even though physical human gestures were, in reality, utilised to control *Middleground* and *Background* phenomena/parameters (and this performance is intended to focus on the control of *Foreground* phenomena/parameters) – the control was of such a contrived, obvious, conventional and non-experimental manner (as well as a necessity, as no music would have been produced), that I feel it does not take away from the credibility of the system as one that explores new and exiting ways to control *Foreground* parameters.

In regards to the controlled *Foreground* parameters of the system/performance, by the nature of the type of parameters under control, it was not possible to compose and put together a phenomena typical of the genre, in the same way it would be for a *Middleground* or *Background* phenomena – such as the establishing of a certain scale or rhythm, or the multiple components of a section (which are all explored in P*erformance 2* and *3* respectively), ready for control. When it comes to *Foreground* phenomena, single parameters are automated in a real-time sense, so the compositional decisions in a musicality sense, in order to produce music of the style, came from the choices of which parameters to manipulate. An infinite amount of possible parameters could have been allocated, but only a few specific ones were selected to be utilised for control, but more importantly (at least in regards to the discussion within this sub-chapter), to be controlled/affected in turn.

Speaking generally, the model of control, which is examined over the chapter, is based around the concept that each instrument has two *Foreground* parameters that, when altered, are in turn used to control two *Foreground* parameters each of the two alternative orchestration layers. The three instruments produce three typical Dubstep layers: *Drums* (performed by reacTable), *Vocals* (performed by vocalist) and *Synth* (performed by hybrid electric guitar set-up). Each layer produces sounds that are fundamentally created within computer software programs (even the *Vocals* run through a computer algorithm before being outputted) – therefore; it was possible to craft this type of parameter-manipulating-parameter control.

The *Drum* layer is produced by the Electronic instrument; the reacTable. In a general sense, the reacTable uses symbol sensing technology to recognise physical objects when placed on the tabletop, generating allocated notes/loops (or any assigned parameter of music), while also being able to track the location, and therefore enabling the X-position, Y-position and rotation of the object to automate allocated parameters of music. In this case, one cube object was utilised so that each face of the cube, when placed face down on the table, would set off a different Dubstep-style drum loop. The hits used to make up the loops were typical electronically processed sounding, and the loops ranged from sparse to denser, including one that was in double-time – all created in Ableton Live. As the nature of the instrument will not generate or manipulate any sounds unless programmed to, I was able to create and embed the two foreground *Controlling Parameters* myself, as opposed to choosing two naturally occurring ones (as will be the case with the two alternative instruments). The two *Controlling Parameters* established were *Distortion amount* and *Low Pass filter position* – assigned to the respective X and Y position of the cube object on the tabletop surface. I will not detail the foreground parameters of the *Synth* and *Vocal* layers that these are used to control here, as that discussion is better suited within the following chapters, including the gesture-to-sound mapping chapter and representative table.

The *Synth* layer is generated by the Hybrid instrument; the Electric guitar. As touched upon, the physical electric guitar is merely used as an interface, as various gesture-produced musical information is sent into a laptop, where processing allows the output to be that of a Synth layer. It is possible to see this instrument in the same way as a MIDI keyboard, however the interface is a guitar, rather than a keyboard. MaxMSP and an application from *Jam Origin* ("Jam Origin", 2014) called MIDI Guitar were necessary components in order to establish the instrument to work as an interface, while it was within Ableton Live that the sounds were produced. As part of the model of control that will be described in the next section, the way this instrument (encompassing the interface as well as computer processing aspects) is designed, it is possible to switch between four tone colours, achieved by playing the notes C, C#, D, D# in the highest register (achieved by utilising certain programming techniques within MaxMSP). These four timbres are indeed discernable, but all still provide the Synth with the characteristic dirty, grungy, distorted synths prevalent in Dubstep music. The two *Controlling Parameters* of this instrument are the switching between the four variable timbres (that is, when a switch is activated, this will also simultaneously activate/control two allocated foreground parameters each of the other two instruments), as well as Amplitude/timbre. Once more, the concept of that second *Controlling Parameter* is a complex one – it is termed "Amplitude/timbre" because within the instrument itself, the natural amplitude changes are used to instead manipulate the a timbral parameter. That is, processes are put in place so that naturally occurring amplitude changes occur due to the interface type (such as from the pluck of a string ringing out), the outputted volume of the synth layer stays constant, but the amplitude information is still extracted and instead is applied to a timbral parameter. In short, as a plucked string rings out, instead of the note becoming softer, a “scream”/“wah” effect amount on the *Synth* orchestration is increased.

The *Vocal* layer is generated by the Traditional instrument; the vocalist (or voice box). Technically the “instrument” also encompasses various computer software, as the vocalist is required to sing into a microphone, where the sonic information is processed within MaxMSP and Ableton Live, and then outputted, with the result including various parameter manipulations determined by the two alternative layers (Drum and Synth). That is, the underlying model of control would not have been possible to apply to a vocal layer without any form of computer processing. Apart from this, the “instrument” is treated in a conventional way in order to produce the layer of sound for the piece/performance (i.e., the singer merely sings). The two *Controlling Parameters* are *Frequency* and *Amplitude*. Once again, these parameters are manipulated from a source in a way that does not step out of the boundaries of the Dubstep style – that is, the singer does not sing, for example, microtonal pitches, or scream at the top of his/her lungs to produce the same exaggerated amplitude every note. The style of the genre remains intact.

3.2.2: Hierarchy 2: Fast moving information

In regards to the second hierarchy – which determines the way in which gestures are interpreted and used for control of musical parameters – the gestures in *Performance 1* have allocated data extracted in a *Fast Moving* manner. Here, I will discuss how data is extracted from certain tracked gestures, interpreted, and then applied to control parameters of music in a live environment. This hierarchy relates to what is seen as part of the performance, and how this aspect is turned into control – it is the “gesture-to” component of the overriding gesture-to-sound relationships within the system.

The phenomena/parameters of control within the system involved in *Performance 1* surround *Frequency*, *Amplitude*, *Timbre* and *Tremelo/Vibrato*. Not only are these the element that are eligible for control, but also the elements in which data will be focused on and extracted from in order to be put towards control. That is, the underlying model of control within this system is the concept of *Foreground* parameters affecting *Foreground* parameters. Each type of control is made possible by different means, but primarily (that is, for the Traditional and Hybrid instrument) utilising methods within MaxMSP to extract *Frequency* and *Amplitude* information. This is then used to further control alternative *Foreground* parameters within the own layer and instrument, in some cases, and furthermore, of other layers' orchestration. Within this chapter, I examine how this model of control was made possible in a general concept sense, and additionally in a specific technology-based sense.

In regards to the *Drum* layer, produced by the Electronic ReacTable instrument – when the *Distortion amount* and *Low pass filter position* parameters are altered by the human instrumentalist, the *Band pass filter position* and *Distortion amount* of the *Vocal* layer, as well as the *Amplitude* and *Reverb amount* of the *Synth* layer, are in turn manipulated. This manipulation is by no feat of the respective *Synth* (Electric guitar) and *Vocal* (vocalist) instrumentalists, as the alterations in the detailed parameters occur as a direct result of the performance decisions made by the *Drum* (reacTable) instrumentalist. This process is presented clearer when represented in a visual table in the *Gesture-to-sound mappings* sub-chapter (See Table 3.2, Table 3.3 and Table 3.4). The control happens in real-time. That is, simultaneously to the *Drum* parameters being altered, the *Synth* and *Vocal* parameters also undergo alterations. Once again, the noted tables detail specific relationships between the parameters (such as the ratio being 1:1, reverse, multiplying, etc). Data extraction in this case is quite simple, as the original *Controlled Parameters* of the source instrument are established within MaxMSP, so these same values (converted to MIDI ranges) are just applied to control the allocated parameters of the Synth and Vocal layers of orchestration.

When it comes to the *Synth* layer, produced by the Hybrid Electric guitar instrument – when the *Timbral sudden changes* are activated, and the *Amplitude/timbre* parameter altered by the instrumentalist, the *Tremelo* and *Reverb amount* of the *Vocal* layer, as well as the *ADSR of allocated drum hit* and *Amplitude* of the *Drum* layer, are in turn activated or manipulated. Once again, this manipulation is all at the hands of the *Synth* instrumentalist. The control once again happens in real-time, and with specific relationships (See Table 3.2, Table 3.3 and Table 3.4) assigned to each mapping. Once again, the data extraction process is not too complicated, as the majority of computer processing for this layer/instrument was detailed in the previous subchapter in regards to the *Timbral sudden changes* and *Amplitude/timbre* elements, which are part of the instrument set-up and not in relation to the model of data acquisition and control in discussion. When it comes to how the gestures determining *Timbral sudden changes* were tracked and using for further control – whenever a change was activated, depending on the option that was activated (remembering there are four), the ADSR (and in particular, only the Attack and Sustain values) of a Snare drum hit of the Drum layer were altered. The relationships being:

Table 3.2: Timbral sudden changes and snare drum relationship

Option / note played to activate	Layer	Parameter	Activated status	
1 / C	Synth	Timbre	Scream value: 0	WT pos. value: 0
	Drums	Snare	Attack: 65	Attack: 65
2 / C#	Synth	Timbre	Scream value: 127	WT pos. value: 127
	Drums	Snare	Attack: 65	Attack: 0
3 / D	Synth	Timbre	Scream value: 0	WT pos. value: 127
	Drum	Snare	Attack: 127	Attack: 65
4 / D#	Synth	Timbre	Scream value: 127	WT pos. value: 0
	Drums	Snare	Attack: 65	Attack: 96

In regards to the *Tremelo* parameter of the *Vocal* layer, the effect merely activates on and off whenever a new *Timbral sudden change* is activated. When it comes to the *Amplitude/timbre* Synth parameter, and how this data is used to control the respective allocated parameters of the *Drum* and *Vocal* layers – as the incoming values from the guitar interface were originally converted from amplitude information into MIDI values, these converted MIDI values are then just applied for further control.

Finally, the third layer of orchestration, being that of *Vocals*, is produced by the Traditional vocalist "instrument". When the *Frequency* and *Amplitude* parameters are altered by the singer, the *High pass filter position* and *Vibrato amount* of the *Synth* layer, as well as the *Reverb amount* and *Frequency of kick-drum hit* of the *Drum* layer, are in turn manipulated. The control takes place simultaneously, with specific assigned relationships between parameters. In regards to data extraction, the singer is required to sing into a microphone[10] in order for information, which is a direct result of physical gestures involved in singing, to be acquired. Here, raw data, represented as two continuous numerical streams – one corresponding to hertz (frequency) and the other to decibels (amplitude) – is converted into MIDI value ranges (i.e., 0-127). These converted values are then applied to control the allocated parameters of the *Synth* and *Drum* layers of orchestration. Again, see Table 3.2, Table 3.3 and Table 3.4 for mapping relationships.

The physical gestures utilised in each case are those of minimal size, and performed almost imperceptively. This makes for new and exciting ways to control and perform live Dubstep music.

3.3: Gesture-to-sound mappings

The following mapping template outlines the gesture-to-sound strategy employed in the provided/documented performance. The tables simply represent the combined information discussed in the previous two subchapters, and can be used as a reference point for when analysing the relevant performance.

The first table outlines parameter control from point of view of each instrumentalist – as in, what they control on their instrument, and in turn what parameter that then controls on the other two layers.

The second table outlines the same information, however, from the point of view of each layer – as in, what parameters are being controlled, and furthermore, where is the control originally coming from.

[10] The microphone use is also necessary in order to process vocal layer with external parameter control manipulation (originally determined by *Drum* and *Synth Controlling Parameters*), as well as to project the overall volume of the layer out of speakers when outputted.

Table 3.3: Colour key

Layer	Colour
Drums	
Vocal	
Synth	

Table 3.4: Performance 1 set-up

Type	Instrument	Layer	Own FG Par. / Controlling Parameters	Aff. FG par.		Relationship
Tradit-ional	Singer	Vocal	Frequency	Synth	HP Filter pos.	Multiplier
				Drums	Reverb amnt.	Multiplier
			Amplitude	Synth	Vibrato amnt.	1:1
				Drums	Frequency of kick-drum hit	Multiplier
Hybrid	Electric Guitar	Synth	Timbral sudden changes	Vocal	Tremolo	1:1 (on/off : on/off)
				Drums	ADSR on Snare Drum	1:1 (on/off : on/off)
			Amplitude/ timbre	Vocal	Reverb amnt.	Multiplier
				Drums	Amplitude	Multiplier
Electro-nic	ReacTable	Drums	Dist. amnt.	Vocal	BP filter pos.	Multiplier
				Synth	Amplitude	Multiplier
			LP filter pos.	Vocal	Dist. amnt.	Reversed
				Synth	Reverb amnt.	Multiplier

Table 3.5: Performance 1 set-up 2

Layer	FG par. utilised	Controlled/ affected by
Synth	Timbral sudden changes	Own
	Amplitude/timbre	Own
	HP Filter pos.	VX – Frequency
	Vibrato amnt.	VX – Amplitude
	Amplitude	DRM – Dist. amnt
	Reverb amnt.	DRM – LP filter pos.
Vocal	Frequency	Own
	Amplitude	Own
	Tremolo on/off	SNTH – timbral sudden changes
	Reverb amnt.	SNTH – Amplitude/timbre
	BP filter pos.	DRM – Dist. amnt.
	Distortion amnt.	DRM – LP filter pos.
Drums	Distortion amnt.	Own
	LP filter pos.	Own
	Reverb amnt.	VX – Frequency
	Frequency of kick-drum hit	VX – Amplitude
	ADSR on snare drum	SNTH – timbral sudden changes
	Amplitude	SNTH – Amplitude/timbre

3.4: Rehearsals

There were no real rehearsals, as the instrumentalists first experience with the system was on the day of recording the demonstration. A read through of the score, and a run through, was all the practice that was had.

3.5: Performance

[Video included on DVD: *Performance 1*]

As can be seen in the video, the "performance" was more of a demonstration, as there was no audience. The participants involved were Tim Cluett on guitar (performing *Synth* layer) and Kaite Campbell on vocals (performing *Vocal* layer). I, myself, was on reacTable (performing *Drum* layer). There was a basic score (see Appendix 2) – but in short, each level was showcased in an allocated section each, and a "jam" section was also included. The system worked as it was supposed to, with no glitches occurring.

3.6: Participant reaction

The participant instrumentalists enjoyed interacting with and experiencing the system – as well as recognising the potential it holds – and they found it fun and interesting to perform in/with. Ease of control, and finding the right combination of parameters in the intertwining system of control to produce a clearer sound were points of suggested improvement, however.

In regards to the use of gestures – even as they were acutely aware of the underlying processes of the interactive music system – neither participant felt that they compromised performative intuition while playing their instrument/singing. The understanding of the system did not create any sense of confusion or apprehension in their approaches to performing. However, the fact that they were conscious that specific gestures they execute would affect allocated musical parameters of all instrumental layers did determine, to an extent, what they performed as well as their expressivity – this, however, was seen as a positive by-product of the system.

3.7: Ways to improve/build – Where to go from here?

I found, on the whole, the system worked well as a starting point to grow from. Refining of how the parameters interact in regards to control is a necessity, however. The layers could have been mixed together better as well, as the outputted sound was muddy, and not overtly clear. The interactivity when jamming was a definite positive, however, and this is the aspect that needs to be focused on and built even more for this system to truly be great.

CHAPTER 4

Evaluation/analysis of system performances: System 2: Dancers

The next performance in the series that aims to insert the human body into the performance of live Dubstep music focuses on the gestures of Dancers. Three dancers control musical phenomena of a length long enough that it is realised in hindsight, by use of medium – yet, full bodied – physical gestures and movement. The use of dancers was important in the context of the entire project, as they are part of an art form that fundamentally revolves around the visual aesthetics that the physicality of the human body can bring. Exploring the medium, and incorporating it into my own project was absolutely necessary. The idea that music can be performed without holding any tangible objects, or instruments involved, was also an appealing concept to explore.

4.1 System set-up/analysis

When it comes to the set-up of *Performance 2*, within the paradigm of the proposed model, three human dancers perform on stage, dancing in a Dubstep style to keep with the aesthetics of the overall performance. There are three types of sensors used in the set-up: a video camera (Playstation 3 EyeCam in conjunction with reacTIVision), MIDI sensors (Miburi suit), and an Xbox Kinect. They are used in different ways to focus on, and collect gestural data, from each dancer individually.

As with the previous analysis of *Performance/System 1*, an overall knowledge of the system and underlying process as a whole needs to be understood in order to fully examine one specific component at a time. So therefore, in a broad sense, three dancers perform on stage in a live setting. The generation of music is not controllable, however, parameters within the piece are. That is, if the dancers stood motionless on stage (depending on various factors, including whether the music had actually initiated), there would still be music produced – this was not the case for the previous performance/system. The dancers stick to their own routines, dancing in unison occasionally, as well as in completely different manners. Various forms of technology, as detailed, are utilised in certain ways in order to control *Rhythm* and *Scale*-based parameters of the respective layers of orchestration of the typical Dubstep piece. That is, each dancer controls their own layer, with no external control possible within the confines of the system. The system, and aligned performance, is interactive in the sense that the dancers are altering the music they are listening to, and in turn reacting to it, and in turn affecting it again. This is on a relatively simultaneous scale too, however, not completely real-time, as the parameters eligible for control are still manipulated in a manner that is in hindsight.

Each dancer controls their own layer by use of their own method of gesture-capturing technology and process. That is:

Dancer 1: Synth (Xbox Kinect)
Dancer 2: Drums (Miburi suit)
Dancer 3: Violin Stabs (reacTivision)

To go into detail about the exact workings of each process of gesture-capturing methodology and the technology specifics involved is outside the scope of this project, however, relevant sources have been cited that provide an overall idea of how they each work (“Xbox 360 + Kinect”, 2013; Vickery, 2002; “reacTIVision 1.4, nd).

Each dancer required a laptop each, as part of the system – running separate MaxMSP patches while sending information through a network to one parent Ableton Live session.

4.2: Application of hierarchies (Level of control 2)

The interactive music system utilised in *Performance 2* applies the *Middleground* level of the *Musical phenomena/parameter* hierarchy in conjunction with the *Medium moving* level of the *Data acquisition/interpretation* hierarchy (See Table 4.1 detailing the system’s overriding level of control).

Table 4.1: Performance 2 level of control

1. Musical phenomena/parameter		2. Data acquisition/interpretation	
Level eligible for control	**Levels not eligible for control**	**Level applied for control**	**Levels not applied for control**
Middleground	Foreground, Background	Medium moving	Fast moving, Slow moving

4.2.1: Hierarchy 1: Middleground phenomena/parameters

In regards to the first hierarchy – which dictates the type of musical phenomena and parameters that are eligible for control – the gestures utilised in *Performance 2* control phenomena/parameters of a *Middleground* level. Here, I discuss what parameters are controlled, as well as boundaries that were set-up to ensure the outputted music remains in a Dubstep style. This hierarchy relates to what is heard in relation to the *Middleground* level of phenomena/parameter categorisation.

Dancer 1 controls the *Middleground* parameters of the *Synth* layer. The sensing technology involved in this process is that of the Xbox Kinect. The device is normally used in conjunction with video games – such as analysing full body movement to control the swing of a golf club – however, in this instance it is utilised to perform live Dubstep music. Generally speaking, the Xbox Kinect can be used to track the X, Y and Z position of any allocated limb. This made the device invaluable in a project like the one undertaken in this study. The discussion about the choice of gestures tracked and used for control is presented in detail below. In regards to the *Synth* layer, and the possible musical changes that can occur, there are numerous parameters eligible for control on this Synth layer. Although, firstly, the ineligible parameters must be established; these parameters lie on the *Foreground* and *Background* layers. In regards to *Background*, initially; for all three layers in this performance (*Synth*, *Drums*, and *Violin Stabs*), *Background* parameters are only realised if the provided score (discussed below) is followed. In saying this, the *Structure* of the piece is non-critical, while, when it comes to the *Texture*, all three layers are always sounding simultaneously throughout the piece. The *Foreground* parameters, however, are more important, with the *Synth* timbre being a standard dirty, grungy, distorted synth, with inherent sub-bass, and a "wobble" assigned to it. When it comes to the eligible parameters for control within this layer, all parameters lie within the realms of *Rhythm* and *Scale* control. The three eligible parameters of control include:

1) Note selection – out of six provided options
2) Switch between the "wobble" LFO being of a sinewave shape or a squarewave shape
3) Speed of "wobble"

The first parameter relates to *Scale*, as even though each note can be chosen, only a select amount of notes (as part of a scale) are provided. The notes are those of a minor pentatonic scale in C. That is, C, D#, F, G, A#, C. The next two parameters relate to *Rhythm*, and more specifically the "wobble" of the synth, and are self-explanatory. When it comes to the speed of the wobble, however, the rate will always be quantised and on the beat. Wobble bass is one of the grey areas between time-scales in music, at least in regards to my project and process of categorising the phenomena. This is because the way the pulse is created in wobble bass is by constantly altering timbral qualities (in this case, amongst other simultaneously manipulated timbral parameters, by altering the high-end frequencies, by use of a Low pass filter). I have classified wobble bass within the rhythmic phenomena categorisation, however, as the main focus is on rhythm. Indeed, timbre is altered, but because the timbral quality is returned to at every "pulse" due to the repeating nature of the allocated waveform enabling the manipulation, the focus lies primarily on the rhythmic aspect.

Dancer 2 controls the *Middleground* parameters of the *Drum* layer. The sensing technology utilised is that of MIDI sensors, and – more specifically – MIDI sensors encompassed in a Miburi suit. Broadly speaking, MIDI sensors are placed on the limbs of a full-bodied suit, and actions made by respective limbs trigger MIDI signals. How I exploited this to control *Middleground* parameters of the *Drum* layer is explained soon enough, however, for now I shall merely detail the eligible parameters. When it comes to the eligible parameters for control within this *Drum* layer, the parameters once again lie within *Rhythm* and *Scale* control. The three eligible parameters of control include:

1) Activation of Drum Loop 1
2) Activation of Drum Loop 2 (one of three variations)
3) Activation of Drum Loop 3

Once again, there needed to be set/fixed *Foreground* parameters – as is the nature of the model of control for *Performance/System 2* – and this requirement dictated that the timbral quality of individual hits were typical sounding, electronically processed drum sounds, characteristic of Dubstep music. The choice of hits and rhythm are what differentiates each of the loops, abiding by the control of *Rhythm* and *Scale* in this performance/system. Loop 1 is a basic beat – sparse and simple; while Loop 3 is a double-time “drop-friendly” beat. There are three variations for Drum Loop 2 – as it would be suggested, each of the three loops are similar, yet comprise of slightly different rhythms and choice of hits.

Dancer 3 controls the *Middleground* parameters of the *Violin Stabs* layer. The sensing technology utilised is that of a Playstation 3 EyeCam working in conjunction with reacTIVision – similar to the underlying technological process of the reacTable. In a general sense, visual glove-sized symbols are placed on the hands (and back) of the dancer, and while she performs physical gestures changing the position of said symbols, the music is altered. The set/fixed parameters include the unchanging timbre of violin samples, as well as no dynamic expression being available. When it comes to the eligible parameters for control within this layer, the parameters once again relate to *Rhythm* and *Scale* control. The three eligible parameters of control include:

1) Chord type
2) Tonic note of chord
3) Notes per bar

In regards to Chord type, there are four possible options:

1) I (single note)
2) I-III-V
3) I-III-VI
4) I-IV-V

In regards to Tonic note of chord, there are three options:

1) C
2) F
3) G

In regards to Notes per bar, there are four options:

1) 1
2) 2
3) 4
4) 8

The notes are spread evenly (as crotchets) and start on the beat.

When it comes to the building of sounds, in regards to each of the three layers, this was all achieved in Ableton Live, as was the arrangement of loops and phrases.

4.2.2: Hierarchy 2: Medium moving information

In regards to the second hierarchy – which determines the way in which gestures are interpreted and used for control of musical parameters – the gestures in *Performance 2* have allocated data extracted in a *Medium Moving* manner. Here, I discuss how data is extracted from certain tracked gestures, interpreted, and then applied to further control parameters of music. This hierarchy relates to what is seen as part of the performance, and how the physicality of the human body is exploited to create music.

In regards to the *Synth* layer, the extraction of gestural data for parameter control is applied in a *Medium Moving* sense in a number of ways. As mentioned, the process associated with the Xbox Kinect is utilised to track the dancer (Dancer 1) responsible for this layer of control, and extract gestural data. Many of the techniques applied were explored in great detail, in a theoretical sense, in the chapter originally outlining this second hierarchy. Firstly, the Y-position of the dancer is observed – the data collected is continuous, however, it is only applied for control when the hand moves into a certain area range. The left-hand is also utilised for control, however, in this case, fast movements, over a certain distance and time must be achieved in order to initiate control. These fast stabbing gestures of left, right, back, forward, up and down all are registered to be used to control different parameters. This also applies to foot-stops, where the left foot is used for control of a parameter, as is the right foot. In both cases, gestures over a smaller than required distance, or at a slower than required speed, will not be registered for control.

When it comes to the *Drum* layer, the extraction of gestural data for parameter control is applied in a way based around the idea of speed of multiple gestures working in tandem. Dancer 2 is wearing a Miburi suit, with attached MIDI sensors (as pictured below) – sharp punching movements of the arms send out MIDI "bangs", and it is the time between these "bangs" that determine the type of control that ensues. There are three levels, which result in three distinct types of sonic result (as will be detailed in the gesture-to-sound mapping chapter). The first level activates control when the time between obvious full arm-length gestures is the length of two beats at 140bpm tempo; the second level activates control when time between the gestures is the length of one beat; and the third level activates control when the time between gestures is the length of half a beat. When the middle length level is activated, there are three sub-levels of possible control activation (regarding setting off of three Loop variations) – each time the middle length (1 beat between gestures) is activated, the three sub-levels of control are run through one after the other. That is, the first time the middle length level is activated, *Loop 2 Variation 1* activates; the second time the middle length level is activated *Loop 2 Variation 2* activates; and the third time the middle length level is activated *Loop 2 Variation 3* activates. Certain programming techniques within MaxMSP were employed to ensure that, if two gestures happened too quickly after one another (perhaps mistakenly by the dancer), they would not register for control – that is, the gestures have to be close enough in sync with the relevant beats in order to enable the parameter control desired.

Figure 4.1: Miburi 1

Figure 4.2: Miburi 2

Figure 4.3: Miburi 3

Figure 4.4: Miburi 4

In regards to the *Violin Stab* layer, the extraction of gestural data for parameter control is primarily based around the idea of hand movements in a two-dimensional plane, activating control of *Middleground* parameters when certain ranges, or areas, are entered into. As touched upon, reacTIVision is utilised to track symbols worn by Dancer 3 – the movement of which is then used to determine sound manipulation. The left hand works on a sole Y-position bases. There are four area ranges the hand/symbol can move into; each one activating a different type of parameter control. The right hand works on the Y-axis also, similarly working between four allocated area ranges, activating a different type of parameter control when each range is entered into. This hand also works on the X-axis too, however, working between three allocated area ranges, activating alternative types of sound manipulation as the hand moves between the ranges. All area ranges in every case are of the same length. The dancer also has a symbol attached to her back, which, when recognised by the symbol-tracking technology (that is, when the dancer turns around and faces her back to the audience), all usual control for this layer stops, and an alternative *Middleground* parameter is controlled/activated – this is clearer in the upcoming *Gesture-to-sound mappings sub*chapter.

4.3: Gesture-to-sound mappings

The following mapping template outlines the gesture-to-sound strategy employed in the provided/documented performance.

Table 4.2: Performance 2 set-up

Dancer/ sensor	Layer	Parameter	Gesture		Sound Result
1 / Kinect	Synth	Rhythm	Right hand Y-position		LFO rate (Speed of wobble)
			Left Foot-stomp		Sinewave shaped wobble
			Right Foot-stomp		Squarewave shaped wobble
		Scale	Fast, sharp hand movement	Left	C1 note ON
				Up	D# note ON
				Down	F note ON
				Forward	G note ON
				Right	A# note ON
				Back	C2note ON
2 / Miburi suit	Drums	Rhythm/Scale	Long time between obvious arm gestures [2 beat length]		Loop 1 activated
			Medium time between obvious arm gestures [1 beat length]		Loop 2 Variation 1 activated
					Loop 2 Variation 2 activated
					Loop 2 Variation 3 activated
			Short time between obvious arm gestures [1/2 beat length]		Loop 3 activated
3 / ReacTIVision	Violin Stabs	Rhythm	Left hand Y-position	Range 1	1 note per bar
				Range 2	2 note per bar
				Range 3	4 note per bar
				Range 4	8 note per bar
		Scale	Right hand X-position	Range 1	C tonic note
				Range 2	F tonic note
				Range 3	G tonic note
			Right hand Y-position	Range 1	I
				Range 2	I-III-V
				Range 3	I-III-VI
				Range 4	I-IV-V
		Rhythm/Scale	Turn around facing back to audience		16 bar melody with complex rhythm/pitch information activated

In regards to the Right hand Y-position of the Dancer 1 Synth layer controlling the LFO rate (Speed of wobble) – as the hand moves from a low to higher position, the speed of the LFO increases.

In regards to the Y-position of either hand of the Dancer 3 Violin Stab layer – Range 1 is the lowest area range, moving upwards to Range 4 being the highest range area, when it comes to hand placement in space.

In regards to the Right hand X-position of the Dancer 3 Violin Stab layer – range 1 is the closest to the body, moving outwards to Range 3 being the furthest from the body, when it comes to hand placement in space.

4.4: Rehearsals

The rehearsal process took place over only 10 days, which, in hindsight, was nowhere near enough time to perfect an operation/performance of this scale. I had only one one-on-one rehearsal with each dancer, and three full run-through rehearsals. Within the one-on-one rehearsals, however, I was able to work closely with the respective dancer, and collaborate on how best to adjust and refine the system in order to allow them to dance in a comfortable and instinctive way, but to also extract the required gestural data to be used for control.

The most difficult part of developing the system was perfecting the networking (See Figure 4.5). Many hours were spent using trial and error to get the laptops talking to each other to enable the performance.

Figure 4.5: Networking set-up

Included on the attached DVD, in the folder entitled *Rehearsals*, are three short videos documenting part of the rehearsal process when it came to this performance/system. The control of the *Synth*, *Drum* and *Violin Stab* layers by the participants, using the relevant technology/gestural-approach, can easily be observed.

4.5: Performance

[Video included on DVD: *Performance 2*]

The performance took place on August 1st 2013 at the Mt. Lawley Edith Cowan University campus Music Auditorium. The dancers involved in the performance were second year dance students from the Western Australian Academy of Performing Arts (WAAPA):

- Melissa Tan as Dancer 1
- Emily King as Dancer 2
- Ezgi Gungor as Dancer 3

A basic and ambiguous set of performance instructions were developed in collaboration with the dancers in order to showcase each level of control, as well as experiment with what could be achieved with the interactive music system:

Table 4.3: Performance 2 instructions

Dancer 1 showcase, while others perform basic moves resulting in 'background' music
Dancer 2 showcase, while others perform basic moves resulting in 'background' music
Dancer 3 showcase, while others perform basic moves resulting in 'background' music
Dance in unison in Dubstep/club style

How this score was interpreted was up to the dancers, as they counted themselves in and out of showcase sections, and decided in rehearsals what type of dance moves they would perform in final unison section.

Additionally, I instructed the dancers to see the first three showcase sections as a way to attempt to "play" the system, with an idea in their head of what sounds would be produced, while in the last section, it was an opportunity to just dance, and see how the system would react – given the various technological, compositional and programming boundaries that were set, the result would always be one of a Dubstep style.

4.6: Participant reaction

Each dancer enjoyed their experience of working in an environment far removed from their comfort zone. Because the interactive music system was set-up to always generate music of a Dubstep style (that was always in time and that always produced harmonising pitch material), this was a reassuring safety net for the dancers; none of whom were well versed in the art of performing music (especially that of experimental electronic music!). They could therefore focus on movement (which is what they do best) rather than trying to perform music.

In regards to the use of gestures, the overriding feeling was that more freedom of movement could be integrated into the system. Specifically, the dancers felt that, although they were relatively unrestricted when it came to limb movements, they were limited in regards to movement around the stage (i.e., position in space), and that this would have improved the performance aesthetically.

4.7: Audience reaction

As the performance was presented in front of first, second and third year Music Composition and Technology WAAPA students, it was a great opportunity to receive immediate feedback within a Q and A forum. I received encouraging comments in admiration of the system set-up, and eager questions on how it was all put together and achieved, however, not all feedback was positive. The main points of criticism I took from the post-performance discussion were that the showcase sections were not clear enough, at times the music sounded "sloppy", and there was not enough variation of and within sections. I noted the feedback given at this performance.

In regards to the use of gestures, once again it was suggested that more freedom could be allocated to the dancers when it came to movement around the stage. However, it was noted that the gestures that *were* utilised – and the way in which they were mapped – encompassed a working balance between obvious gestures that were evident immediately, and not-so-obvious gestures, which added intrigue and interest to the performance.

4.8: Ways to improve/build – Where to go from here?

I believe the performance was a success in terms of a starting point, and having established a solid basis for new and exciting ways to perform Dubstep music. I agree that the above detailed criticisms should be addressed, however. A next logical step would be to allow the dancers more freedom (and not just in regards to movement around the stage), as they had restrictions placed on them in the sense that they had to move in very specific ways in some cases. Refinements in the system to allow greater musical variation would also be something to look into.

CHAPTER 5

Evaluation/analysis of System performances: System 3: Audience members

The third performance differs from the previous two discussed, in the way that, in the first two, the bodies of the artists are embedded into the performance, while in the third performance it is in fact the bodies of the audience members as a whole that are utilised to insert a human presence element into the performance of Dubstep music. Although in *Performance 2*, the human body – without the use of physical instruments – was used to generate and control the music, more detailed movement from the one body manipulated more precise musical parameters. In *Performance 3*, however, broader musical parameters are manipulated from audiences' bodies utilised as a single object. The interactive music system is set up so that the collective movement and behavior of the audience bodies control certain aspects of the music produced. This idea relates to the concept that audiences play a major role in the performances of contemporary electronic music, discussed by electronic musicians such as Ben Neil:

> One of the key ideas to come out of recent electronic pop culture is the 'rave' sensibility in which the traditional notions of performer and audience are completely erased and redefined. In this type of event, the artists are not the center of attention; instead it is the role of the artist to channel the energy of the crowd and create the proper backdrop for their social interaction. The audience truly becomes the performance. (Neill, 2002, p. 4)

Ostertag also states, when discussing the human body's presence in contemporary electronic music performance:

> The bodies are the audience's, not those of the performers. So the physical bond of performance is that everyone is dancing, while the performers hide behind a light show or a fog machine. (Ostertag, 2002, p. 12)

Joel Chadabe also states that, by being predictable, “popular music conveys the sense to a public that ‘you’ (the audience) could have composed it” (Chadabe, 2000, p. 10). The aim of the interactive system I have developed is to take this idea a step further so that in fact the audience is, in a sense, continuously composing the work (or at least aspects of it) as it is being performed. This idea has been explored by Ryan Ulyate and David Bianciardi, who created the *Interactive Dance Club* (Ulyate & Bianciardi, 2002), and incorporated breaks in light beams, heat tracking infrared cameras and MIDI trigger pads (amongst other methods), interacted with by the audience, to control parameters of the music. At this stage in my research, I have not investigated extensively the possibilities of this type of interactive set-up, however, I do wish to explore similar concepts that are employed in the *Interactive Dance Club*. I did, however, want the audience to be acting as an audience traditionally acts at a contemporary electronic music or Dubstep performance, that is, standing up rather than sitting down, jumping up and down, moving around the entire dance floor, dancing, clapping, and throwing their hands up in the air, and not intentionally performing ‘out-of-place’ actions in order to control the music. I then used these collective actions to generate certain aspects of the music and control certain parameters, once again incorporating the idea of a hierarchy.

In order for the concept to work, the audience needed to ‘believe’ it was at an actual performance, which is why I will attempted to create this environment in various ways, which I detail shortly.

5.1: System set-up/analysis

The main objective of *Interactive Music System 3* is to insert the physicality of the human body into live Dubstep performance, by empowering members of the audience with control over certain musical aspects. Various forms of technology, used in conjunction with complex software algorithms, as well as an arduous setting up process were all necessary to create the system and experience, and achieve the final outcome. The tangible result, however, is that audience members’ gestures and movement play a part in performing the music they are listening and dancing to.

When it comes to analysing a system and performance of this nature, an overall knowledge of the system and underlying process as a whole needs to be understood in order to fully examine each contributing element separately. So therefore, speaking in a general sense, the audience members are asked to put on the choice of a red, yellow, blue or green coloured t-shirt. The music is initiated, and as the audience dance and move around the square dance floor, colour-tracking technology is utilised in certain ways in order to control the *Texture* and *Structure* of a typical Dubstep piece. It is clear that the interactive music system, and aligned performance, is interactive in the sense that, to an extent, the audience are determining the music they hear; they proceed to react to that music by moving and dancing; and their collective movements continue to control the music. This is discussed in more detail below.

5.2: Application of hierarchies (Level of control 3)

The interactive music system utilised in *Performance 3* applies the *Background* level of the *Musical phenomena/parameter* hierarchy in conjunction with the *Slow moving* level of the *Data acquisition/interpretation* hierarchy (See Table 5.1 detailing the system's overriding level of control).

Table 5.1: Performance 3 level of control

1. Musical phenomena/parameter		2. Data acquisition/interpretation	
Level eligible for control	**Levels not eligible for control**	**Level applied for control**	**Levels not applied for control**
Background	Foreground, Middleground	Slow moving	Fast moving, Medium moving

5.2.1: Hierarchy 1: Background phenomena/parameters

In regards to the first hierarchy – which dictates the type of musical phenomena and parameters that are eligible for control – the gestures utilised in *Performance 3* control phenomena/parameters of a *Background* level. Here, I do not discuss how the parameters are controlled, but merely *what* parameters are controllable, and the model of control in which they exist, as well as compositional decisions that were made in a musicality sense. This hierarchy relates to what is heard as part of the performance, and is the "sound" component of the overriding gesture-to-sound relationships within the system.

As mentioned, the *Background* phenomena/parameters eligible for control within the system involved in *Performance 3* are *Layer* and *Structure*. In order to establish boundaries of control, in regards to *Layer* and *Structure*, many musical elements had to be pre-determined and set, or fixed – these consist of the phenomena/parameters of the *Foreground* and *Middleground* levels. The basic layout of the piece is made up of four layers – *Synth*, *Drums*, *Bass*, and *Whitenoise*. This instrumentation is typical of a Dubstep piece, and each of the layers were created on their own tracks in the Digital Audio Workstation Ableton Live, using the software's own instruments and effects, along with third party plug-ins. Six sections, as part of an overall structure, were composed independently, where that, if all four layers were continuously playing/sounding, the musicality would work (within the confinements of a typical Dubstep piece) rhythmically, as well as tonally.

When it comes to the six sections, as part of the overall structure, Dubstep characteristis were abided by in the compositional process. The six sections include *Intro*, *Build Up 1*, *Drop 1*, *Build Up 2*, *Drop 2*, and *Outro*. Features such as double-time drum patterns were applied in the Drop sections, and well as the use of simplistic loop-based melody and phrases, and the emphasis on a pulse-oriented feel – primarily showcased by use of violin stabs. The Bass and Whitenoise loops, as a point of interest and reference, only have sound activated within one half of the loop as a whole. The tempo of the entire work sits at a constant 140bpm.

Compositional decisions had to be made concerning the boundaries created within the layering and structuring of the piece. Certain elements were suitably chosen for live manipulation, while deliberate compositional techniques were employed in order to ensure that when parameter manipulation did occur, the transitions – or musical alterations – did not seem out of place, and furthermore, the Dubstep characteristics of the music remained intact.

In a way, the system exists *as* a performance of a single Dubstep piece of music. That is, the piece is initiated, and control by the bodies of audience members takes part from there. In regards to *Structure* control, the six sections – as part of an overall structure – lead progressively into each other, and the audience bodies' gestures are utilised (in ways I will discuss shortly) to control the length of each section – being 16, 24 or 32 bars – (The model of control switches up within the Drop sections, but this is all revealed in time). To make this control effective, two objectives need to be achieved; the first is to ensure an obvious change over of sections to listeners, while the second is to make sure the transitions between sections are seamless and don't seem out of place. The first aim is attained by making each section as distinct and discerning as possible, and applying compositional techniques discussed in the *Features of Dubstep* subchapter to make each section typical of equivalent sections in a piece of the genre. The second aim is attained by applying certain compositional techniques, such as drum fills and tonal cadences, so that, whether the section is controlled (in the live environment) to truncate after bar 16, 24 or 32, the transition between sections seems meant-to-be, as well as characteristic of the genre. That is to say, each section was composed to be 32 bars, even though, by nature of the system and underlying process of control, not every section controlled in this way (if any) will play for the entire 32 bars during performance.

The four layers – *Bass*, *Drums*, *Synth* and *Whitenoise* – were all composed within Ableton live, using built in instruments and effects, as well as third-party plug-ins. In a general sense, the *Drum* layer was comprised of typical electronically processed hits; the *Bass* layer was created using a deep sine-wave subbass sample; the *Whitenoise* layer was a high-end "unpitched" noise sample; and the *Synth* layer – and it's encompassed sub-layers (which is examined shortly) – was created using various compositional techniques, included the chopping up of bounced down originally synthesised synth layers, sampling of traditional instruments sounds, as well as sounds created from scratch using numerous synthesisers within Ableton Live.

When it comes to the manipulation of *Layer* aspects – in particular the compositional and musicality aspects – the model of control is centered around four layers of varying instrumentation, being *Drums*, *Synth*, *Bass* and *Whitenoise*. The audience bodies' gestures are utilised (in ways I eventually examine) to control which layers are sonically present or not. This control takes place on a four-bar by four-bar basis. Certain boundaries within the model of control are set; for one, the Synth layer is treated differently to the other three. Without going into detail, the Synth layer is made up of three distinct sub-layers; 1) a dirty, grungy, wobble bass synth sound, 2) a pad sound, with a long amplitude attack and release, and 3) an arpeggiated violin-stab sound, that follows the chords of the pad sound. The three sub-layers were necessary, as the overall *Synth* layer is controlled in a way that it could never be completely inaudible – instead, at least two of the three sub-synths would always be audible. This is because, without the *Synth* layer, even with the remining three layers audible, the music sounds sparse and out of character for a Dubstep piece.

Layer control – and by extension, the compositional process in a musicality sense – within the two *Drop* sections are also treated differently. The Synth layer changes to include only two sub-synths, which are both comprised of dirty, grungy wobble-bass synth sounds. The method of control entails the audible interchanging between the two layers, once again on a four-bar by four-bar basis. In reality, it is not so much two sub-synths as it is two tracks, audibly changing between each other – the synth phrases on each track, additionally, are created using various characteristic Dubstep compositional techniques, such as applied stuttering, panning, and beat repeat. The remaining layers are ineligible for control, however, still have a presence during the *Drop* sections.

5.2.2: Hierarchy 2: Slow moving information

In regards to the second hierarchy – which determines the way in which gestures are interpreted and used for control of musical parameters – the gestures in *Performance 3* have allocated data extracted in a *Slow Moving* manner. Here, I discuss how data is extracted from certain tracked gestures, interpreted, and then applied to control parameters of music in a live environment. This hierarchy relates to what is seen as part of the performance, and how this aspect is turned into control - it is the "gesture-to" component of the overriding gesture-to-sound relationships within the system.

As already noted, the two elements of control within the system involved in *Performance 3* surround *Structure* and *Layer*. Both are controlled in different ways, but primarily using colour-tracking technology to focus on the movement of audience members, who are asked to don a provided colour t-shirt of Red, Blue, Green or Yellow. As the audience move and dance (gestures of large size and weight), their actions are responsible for the manipulation of certain *Structure* and *Layer* parameters of the outputted music. Within this chapter, I examine how this model of control was made possible in a general concept sense, and additionally in a specific technology-based sense.

When it comes to the data acquisition and control *Slow-Moving*-method applied to manipulate *Structure* parameters, numerous processes were put into place. Firstly, in regards to *Layer* control, each colour of the four t-shirts are utilised to control their own allocated layer:

> <u>Red</u> controls <u>Synth</u>
> <u>Blue</u> controls <u>Bass</u>
> <u>Green</u> controls <u>Drums</u>
> <u>Yellow</u> controls <u>Whitenoise</u>

A Playstation 3 EyeCam is set up to visually capture information, looking downwards onto the allocated dance-floor in which the audience members dance and move around. This video of a 2d plain is sent into MaxMSP in real-time for further processing, and to where the crux of the gestural interpretation is undergone. The general collective position of all the shirts of an allocated colour is tracked, and the average position is registered. That is, a single value on the horizontal scale, as well as a single value on the vertical scale, is produced. At the very start of every four bars, a snapshot of values is recorded, and registered for control of the relevant layer's audibility to be applied. In regards to the *Blue* (controlling the *Bass* layer), *Green* (controlling the *Drum* layer), and *Yellow* (controlling the *Whitenoise* layer), when the horizontal value is greater than the vertical value – and the snapshot of values is taken at this time to be used for control – the linked layer is manipulated to be audible. Whereas, at the time of the snapshot of values being taken for control, if the horizontal is less than the vertical, then the layer is manipulated to be inaudible. In both cases, the time of in/audibility lasts for four bars, before a new snapshot of values is taken and applied.

In regards to the Red shirts, which control the *Synth* layer, the method of control is slightly varied. As the *Synth* layer is made up of three sub-synths, the method of control cannot be confined to merely an ON/OFF model. As previously touched upon, the *Synth* layer will never be completely inaudible; instead, the established boundaries of control comprise of different combinations of the three sub-synths. Therefore, there are four possible options, in regards to the *Synth* layer, which the movement of the Red-shirt-wearing audience members manipulate in the live performance setting:

<u>Options:</u>

1) [Sub-synth 1 + Sub-synth 2]
2) [Sub-synth 1 + Sub-synth 3]
3) [Sub-synth 2 + Sub-synth 3]
4) [Sub-synth 1 + Sub-synth 2 + Sub-synth 3]

In regards to the how these are controlled, and which combination is audible, the same tracking and data extraction method is applied. That is, the collective position of all red shirts produce a single average horizontal value, as well as a single vertical value. At the time when the snapshot is taken, if the vertical value is greater than the horizontal value, option 1 is enabled. However, if the horizontal value is greater than the vertical value, options 2, 3 and then 4 are progressively run through and activated. Once again, the time each option is in effect lasts for four bars, before a new snapshot of values is taken and applied.

The above model of control, in regards to each specific colour and layer, is in use during all sections – and in constant operation – except for the period of the two *Drop* sections. As the *Drop* section of any Dubstep piece is widely regarded as the focal point, a conscious decision was made to disable control of the length of the two *Drop* sections, so that both would last the entire 32 bars. The type of control otherwise utilised to control section length (which is detailed shortly), was adopted to control *Layers* within the two *Drop* sections. As is the importance of the two *Drops*, it was paramount that the *Drum* and *Bass* layers were present, and the possibility that they could be inaudible during four-bar divisions of the *Drops* had to be eradicated. So in saying this, the method of control where each coloured t-shirt controls the audibility of their allocated layers was also disabled. This is how it became possible that the collective position of all coloured shirts were able to control the Synth *Layer* within the *Drop* sections – which, during these sections, is the only parameter eligible for control. As detailed previously, two tracks were created of two *Synth* layers of a typical Dubstep drop – let them be known at *DropTrack1* and *DropTrack2*. During the *Drop* section both tracks play through simultaneously, however, the audibility of one over the other is controlled by the movement and positioning of all audience members. More specifically, if the sum of each of the tracked coloured shirt's horizontal values is greater than the sum of each of the tracked coloured shirt's vertical values at the time the snapshot of values is taken for control, *DropTrack1* became audible; while if the sum of the vertical values are greater than the sum of the horizontal values, *DropTrack2* became audible. Once again, this kind of data acquisition and control takes place over a four-bar by four-bar basis.

When it comes to the control of structure, a similar method of gestural data extraction and control is employed. The movement of all audience members, regardless of what coloured t-shirt they are wearing, is utilised to determine whether each section (apart from the *Drop* sections in which this type of control is disabled – so therefore, *Intro*, *Build Up 1*, *Build up 2* and *Outro*) would last for 16, 24 or 32 bars. Like the previously described method of *Layer* control during the *Drop* sections, the collective position of all coloured t-shirts provides the data used for control, however, the way in which this data is interpreted is primarily different. The snapshot of values is taken on the very first beat of the first bar of each new section, and these values are further used to determine the length of that section. How these values are interpreted is determined by an algorithmic process within MaxMSP – firstly, the sum of each of the tracked coloured t-shirt's horizontal values is divided by the sum of each of the tracked coloured t-shirt's vertical values; in terms of control, if the value, at the time of the snapshot, sits within the range of 0.6 – 0.8, the relevant section lasts for 16 bars; if the value sits within the range of 0.8 – 1.0, the section lasts for 24 bars; and if the value sits within any other range, the relevant section lasts for 32 bars. As has been described, this model of control of Structure – and more specifically, the length of each section – is disabled during the *Drop* sections, while exclusive *Layer* control takes over.

Regarding software management, certain tactics were used in order to minimise risk during performance. For example, all instrumentation layers were created in separate tracks – many of which relied on multiple plug-in effects. Because the CPU was of a significant amount, with all tracks enabled during performance, each were separately bounced down to audio tracks, which decreased the CPU usage considerably. This was necessary, in a purely reliability sense, to minimise risk of the software – and by extension, the system – crashing during performance. From here, *Layer* control was merely the automation of the volume amount on each track, while *Structure* control was the setting off of 6 scenes (collections of clips) representing each section.

5.3: Gesture-to-sound mappings

The following mapping template outlines the gesture-to-sound strategy employed in the provided/documented performance. The tables simply represent the combined information discussed in the previous two subchapters, and can be used as a reference point for when analysing the relevant performance. The tables represent what happens when the snapshot of values is taken, and is registered for parameter control.

The first table represents the standard model of control, while the second table details the model of control utilised within the two *Drop* sections.

Table 5.2: Performance 3 set-up

<table>
<tr><th rowspan="2">Parameter</th><th colspan="3">Gesture</th><th rowspan="2">Sound Result</th></tr>
<tr><th>Physical</th><th colspan="2">Data extracted/interpreted</th></tr>
<tr><td rowspan="11">Audibility of Layer (Texture)</td><td rowspan="2">Green t-shirt collective position</td><td colspan="2">Horizontal value > Vertical value</td><td>Drum ON</td></tr>
<tr><td colspan="2">Vertical value > Horizontal value</td><td>Drum OFF</td></tr>
<tr><td rowspan="2">Blue t-shirt collective position</td><td colspan="2">Horizontal value > Vertical value</td><td>Bass ON</td></tr>
<tr><td colspan="2">Vertical value > Horizontal value</td><td>Bass OFF</td></tr>
<tr><td rowspan="2">Yellow t-shirt collective position</td><td colspan="2">Horizontal value > Vertical value</td><td>Whitenoise ON</td></tr>
<tr><td colspan="2">Vertical value > Horizontal value</td><td>Whitenoise OFF</td></tr>
<tr><td rowspan="4">Red t-shirt collective position</td><td rowspan="3">Horizontal value > Vertical value</td><td>Option 1</td><td>Sub-synth 1 + Sub-synth 3</td></tr>
<tr><td>Option 2</td><td>Sub-synth 2 + Sub-synth 3</td></tr>
<tr><td>Option 3</td><td>Sub-synth 1 + Sub-synth 2 + Sub-synth 3</td></tr>
<tr><td colspan="2">Vertical value > Horizontal value</td><td>Sub-synth 1 + Sub-synth 2</td></tr>
<tr><td rowspan="3">Structure – Length of section</td><td rowspan="3">All coloured t-shirts’ collective position</td><td rowspan="3">Sum of horizontal values / sum of vertical values</td><td>0.6 – 0.8</td><td>16 bars</td></tr>
<tr><td>0.8 – 1.0</td><td>24 bars</td></tr>
<tr><td>Any value ≠ 0.6 – 1.0</td><td>32 bars</td></tr>
</table>

Table 5.3: Control within Drop sections

<table>
<tr><th rowspan="2">Parameter</th><th colspan="2">Gesture</th><th rowspan="2">Sound Result</th></tr>
<tr><th>Physical</th><th>Data extracted/interpreted</th></tr>
<tr><td rowspan="2">Audibility of Layer (Texture)</td><td rowspan="2">All coloured t-shirts’ collective position</td><td>Sum of horizontal values > sum of vertical values</td><td>DropTrack1</td></tr>
<tr><td>Sum of vertical values > sum of horizontal values</td><td>DropTrack2</td></tr>
</table>

5.4: Rehearsals

As the first human trials would take place during the performance – due to the difficult nature involved in organising upwards of 20 participants – some form of alternative rehearsal process was required. As the method of data acquisition and control was based around colour tracking that was applied to a captured standard-dimensioned video, I would have to somehow emulate what the Playstation 3 Eyecam would be captured during performance. To do this, I created a patch inside MaxMSP of a video in which 20 circles of the same shape, but different colours (Red, Green, Blue and Yellow) moved in a random manner over a white background. From this video, I could apply and refine the colour tracking techniques that I would eventually utilise in the performance.

5.5: Performance

[Video included on DVD: *Performance 3*]

The performance took place on April 5th 2013 at 7.30pm at Mt Lawley Edith Cowan University campus as part of a Music Composition and Technology concert entitled *Spectrum*. As is evident from the video, the system was a success, and multiple audience members participated by donning the coloured t-shirts to be a part of the controlling of live Dubstep music. The system worked without a glitch, and a variety of layer combinations and section lengths were generated.

5.6: Participant/audience reaction

When audience members knew they were controlling parts of the music, they became more enthusiastic about being involved. They tried dance moves, moved to different parts of space, and tried to control the system themselves. Many stuck around to enquire about how system works afterwards. The collective reaction was overwhelmingly positive; and in this respect, made this performance the most successful of the three.

In regards to the use of gestures, it was proposed that more Dubstep-style ‘dance’ moves could be utlised – as it was purely the position in space of collective bodies that was manipulating the music. This concept could still be achieved while abiding to ‘large, collective gestures’ controlling musical parameters (as is this case within this system’s model of hierarchy) – as one suggestion was that the jumping up and down (with optional fist-pumping) of audience members could be exploited – a move seen on club dance floors around the world.

5.7: Ways to improve/build – Where to go from here?

Even though audience movement is directly responsible for changes in parameters of music, it still seems quite hard for audience to realise what they are controlling – to remedy this, a rethink of what gestures are used to make these changes needs to occur. (As well as the recommendations detailed above…) T-shirts are relatively large in surface area, and it is also not practical in club environment to ask patrons to wear specific coloured clothing. The idea of narrowing down the model of control to track more practical coloured objects, like glow-sticks, could be explored.

CHAPTER 6

Future work: Towards a fully interactive system

The logical final step would be to combine the two hierarchies (and all encompassed levels) into the one interactive music system - whereas all parameters of music are eligible for control, by way of instrumentalists, dancers and audience members during the one performance. This type of system was outside the scope of my Masters by Research project, unfortunately. In working to develop a fully interactive system, however, already, many hurdles to overcome can be foreseen. For example, the instrumentalists would need some form of digital scores sent to them in a live and real-time manner for them to abide by the Middleground and Background parameters they must perform. That is, due to the fact that at the start of each performance, these MG and BG parameters would be unknown. In saying that, the prospect of a fully interactive system is an exciting one, and one that would enable truly new and original ways of performing live Dubstep music by inserting the physicality of the human body to an even greater extent

Conclusion

Dubstep is a recent genre in the early stages of its place in contemporary electronic musical history. It will surely develop musically, as well as performatively; however, at the moment the majority of performances, along with equipment and instruments used, are very formulated and predictable. In order to make progress in this area, deviation from the norm is required. The project I have undertaken addresses the issue of a lack of physicality apparent in the performance practice of Dubstp by embedding a human element.

In an endeavor to explore alternative and effective ways of performing Dubstep music, three interactive music systems were developed. Two performances and one demonstration were presented and documented in order to showcase each system. The performances and demonstration were, in essence, experimental, and should be considered as such – as starting points to be built on. That is to say, the musical quality of the performances are not yet sufficient for club performance, which is the likely environment for performances of this nature to ultimately be presented at. In saying this, the technical and performative insights gained through the experience of this project were invaluable. For one, the rigorous procedures (as well as the know-how) required to set-up the equipment and technical side of each system were all-important lessons in themselves – let alone then having to be able to explain the underlying and constant processes in place in layman's terms to the participants, and furthermore, to then take on their suggestive feedback and refine the system to adhere to the requested conceptual modifications – a task that was often easier said than done (i.e., to meticulously modify complex computer algorithms to work in conjunction with multiple forms of hardware in a way that will carry out the intended function successfully, all the while encompassing and maintaining a crucial margin for human error and inexactness). However, this obligation was not a blight on the experience or success of the project. In fact, it was the opposite; the interdisciplinary nature of the study necessitated that alternative approaches be considered, and that methods typically relied upon be compromised, in order to achieve a set of original performance systems that still managed to produce music confined to the parameters of the Dubstep genre. It was imperative to retain the intrinsic traits of the other disciplines:

Performance 1: Instrumentalists and singers brought the energy and intuitiveness of a more traditional style of music performance that is unapparent in live electronic music, with it's seemingly inherent use of finger-based instruments that limit expressive music making gestures.

Performance 2: Dancers (participants of a traditionally non-music making art-form) were utlised as both contributors to musical change, and as visual human reactants – simultaneously altering and adapting to the music through movement.

Performance 3: The audience was acting exactly as you would expect an audience at a Dubstep concert to (i.e., constantly moving around, quick movements – confined to the one space or 'dance floor'), all the while, unknowingly, yet undeniably, determining the music they were listening and dancing to.

I hope the experience and knowledge gained from this project will lead – even if only in my own practice – to new ways of composing and performing music within the boundaries of the genre of Dubstep music.

References

Ableton Live (Version 8). (2013). Retrieved from https://www.ableton.com/en/shop/

All music: Dubstep. (2011) Retrieved from http://www.allmusic.com/explore/style/d13813

Armstrong, T. (2013). Disclosure Unplugged: Why DJs Need To Step Down From The Stage. Retrieved from http://sabotagetimes.com/music/disclosure-unplugged-why-djs-need-to-step-down-from-the-stage/

Bongers, B. (2000). Physical interfaces in the electronic arts: Interaction theory and interfacing techniques for real-time performance In M. M. W. a. M. Battier (Ed.), *Trends in Gestural Control of Music* (pp. 41 - 70): Ircam - Centre Pompidou.

Bongers, B. (2006). *Interactivation: Towards and e-cology of people, out technological environment, and the arts*. Ph.D. thesis, Vrije Universiteit Amsterdam.

Cascone, K. (2003). Grain, sequence, system: Three levels of reception in the performance of laptop music. *Computer Music Journal, 22*(4), 101-104.

Chadabe, J. (1997). *Electric sound: The past and promise of electronic music*. Upper Saddle River, NJ: Prentice-Hell, Inc

Chadabe, J. (2000). Remarks on computer music culture. *Computer Music Journal, 24*(4), 9-11.

Chaîne de FckFakeDJ. (2011). Proof that David Guetta is a fake DJ. Retrieved from https://www.youtube.com/watch?v=HjVecHSL4Rg

Clark, M. (2007). Grime/Dubstep Retrieved from http://pitchfork.com/features/grime-dubstep/6614-grime-dubstep/

Clark, M. (2009). The year in grime/dubstep: The year in dubstep, grime and funky 2009 Retrieved from http://pitchfork.com/features/grime-dubstep/7728-grime-dubstep/

Collins, N. (2003). Generative music and laptop performance. *Contemporary Music Review, 22*(4), 67-70.

d'Escrivan, J. (2006). To sing the body electric: Instruments and effort in the performance of electronic music. *Contemporary Music Review, 25*(1), 183-191.

Drummond, J. (2009). Understanding interactive systems. *Organised Sound, 14*(2), 124-133.

Eigenfeldt, A., Philippe, P. (2011) Towards a generative electronica: Human-informed machine transcription and analysis in Max/MSP.

ELI5: How do people "perform" dubstep at a concert? (2013). Retrieved from http://www.reddit.com/r/explainlikeimfive/comments/1namgr/eli5_how_do_people_perform_dubstep_at_a_concert/

Gizmodo: All DJs Are Glorified Button Pushers (nd). Retrieved from http://gizmodo.com/5921053/all-djs-are-glorified-button-pushers

Golden, E. Is Planning a DJ set Cheating? (2012). Retrieved from http://www.djtechtools.com/2012/03/19/is-planning-a-dj-set-cheating/

Gritten, A. & King, E. (Editors) *Music and Gesture*. Ashgate, Hampshire, 2006.

Hunt, A. & Kirk, R. 2000. Mapping Strategies for Musical Performance. In M. M. Wanderley and M. Battier (eds.) *Trends in Gestural Control of Music*. Paris: IRCAM–Centre Pompidou.

Hunt, A., Wanderley, M. & Paradis, M. (2002). The importance of parameter mapping in electronic instrument design. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-2002)*. Dublin, Ireland.

Jam Origin: MIDI Guitar. (2014). Retrieved from http://jamorigin.com/products/midi-guitar/

Jensenius, A. (2014). To gesture or not? An analysis of terminology in NIME proceedings 2001-2013. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 217-220.

Jessop, E. N. (2010). *A gestural media framework: Tools for expressive gesture recognition and mapping in rehearsal and performance* (Doctoral dissertation, Massachusetts Institute of Technology).

John Cage & Merce Cunningham — Variations V (nd). Retrieved from http://blogs.dharma.art.br/2011/11/john-cage-merce-cunningham-—-variations-v/

Jorda, S. *Digital Lutherie: Crafting musical computers for new musics' performance and improvisation*. PhD Dissertation, University of Pompeu Fabra, Barcelona.

Justice Fake DJ Sets: Mixer Unplugged During 'Live' Set? (2008). Retrieved from http://www.ilovedjme.com/index.php/justice-fake-dj-sets-wtf/

Keith, S. (2010). *Bridging the gap: Thoughts on computer music and contemporary (popular) electronic music*. Paper presented at the Australasian Computer Music Conference, Canberra, AUS.

Kiefer, C., Collins, N., & Fitzpatrick, G. (2008). Evaluating the wiimote as a musical controller. In *Proceedings of the 2008 International Computer Music Conference (ICMC)*.

McKinnon, M. (2007). South London calling Retrieved from http://www.cbc.ca/arts/music/dub_style.html

Miller, L. E. (2001). Cage, Cunningham, and Collaborators: The Odyssey of Variations V. *The Musical Quarterly*, 85(3), 545-567.

Miranda, E. R., & Wanderley, M. M. (2006). *New digital musical instruments: Control and interaction beyond the keyboard* (Vol. 21). Middleton, WI: A-R Editions.

Mulder, A. (1994). Human movement tracking technology. Burnaby, B.C., Canada: Simon Fraser University.

Native Instruments. (2008). Richie Hawtin on his Traktor Scratch DJ setup - Pt 1. Retrieved from https://www.youtube.com/watch?v=J1H9W_iLff0

Neill, B. (2002). Pleasure beats: Rhythm and the aesthetics of current electronic music. *Leonardo Music Journal, 12*, 3-6.

Ng, K. (2004). *Music via motion: Transdomain mapping of motion and sound for interactive performances.* Paper presented at the IEEE.

Odowichuk, G., Trail, S., Driessen, P., Nie, W., & Page, W. (2011) Sensor Fusion: Towards a Fully Expressive 3D Music Control Interface. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim)*, pages 836–841.

Ostertag, B. (2002). Human bodies, computer music. *Leonardo Music Journal, 12*, 11-14.

Overholt, D., Thompson, J., & Putnam, L. (2009). A multimodal system for gesture recognition in interactive music performance. Computer Music Journal, 33(4), 69 - 82.

Pressing, J. (1993). Relations between musical and scientific properties of time. *Contemporary Music Review*, *7*(2), 105-122.

Puckette, M. (2014). Max/MSP (Version 6): Cycling '74. Retrieved from http://cycling74.com/products/max/

reacTIVision 1.4: a toolkit for tangible multi-touch surfaces. (nd). Retrieved from http://reactivision.sourceforge.net/.

Roads, C. (2001). *Microsound*. Cambridge, MA: MIT Press.

Rowe, R. (1993). *Interactive music systems*. Cambridge, MA: The MIT Press.

Sande, K. (2007). Dubstep 101 Retrieved from http://www.residentadvisor.net/feature.aspx?822

Skrillex live on stage (nd). Retrieved from http://www.xsens.com/en/entertainment/live-entertainment/skrillex-live-on-stage

Stuart, C. (2003). The object of performance: Aural performativity in contemporary laptop music. Contemporary Music Review, 22(4), 59-65.

Turner, T. (2003). The resonance of the cubicle: Laptop performance in post-digital musics. *Contemporary Music Review, 22*(4), 81-92.

UKF Dubstep. (2014) Retrieved from https://www.youtube.com/user/ukfdubstep?ob=4&feature=results_main

Ulyate, R., & Bianciardi, D. (2002). The interactive dance club: Avoiding chaos in a multi-participant environment. *Computer Music Journal, 26*(3), 40-49.

Vickery, L. (2002). The Yamaha MIBURI MIDI Jump Suit as a Controller for STEIM's Interactive Video Software Image/ine. Proc. *Australian Computer Music Conference.*

Wii (2013). Retrieved from http://www.nintendo.com/wii

Winckel, F. (1967). *Music, Sound, and Sensation*. New York: Dover Publications.

Winkler, T. (1995). *Making motion musical: Gesture mapping strategies for interactive computer music.* Paper presented at the International Computer Music Conference.

Winkler, T. (1998). *Composing interactive music: Techniques and ideas using Max*. Cambridge, MA: The MIT Press.

Xbox 360 + Kinect (2013). Retrieved from http://www.xbox.com/en-AU/Kinect

Appendix 1

Australian Computer Music Conference 2011 Paper

MUSICAL PARAMETER MANIPULATION POSSIBILITIES OF A HOMEMADE REACTABLE

James Herrington
Masters by Research Student

Lindsay Vickery
Head of Composition

Western Australian Academy of Performing Arts
Edith Cowan University
2 Bradford St.
Mt. Lawley WA 6050

ABSTRACT

Musical parameter control is an important part of live interactive electronic computer music. Due to the increasing availability and affordability of music technology, including powerful computer software, advances in this area are being made to enable easier and more effective parameter control.

The purpose of this paper is to investigate and discuss the musical parameter manipulation possibilities of a homemade instrument with a tangible tabletop interface based on the technology of the reacTable. The design and construction of the instrument is documented, including the physical build as well as the software component of the system, which incorporates the computer software *ReacTIVision*, *Max/MSP* and *Reason*. The core of the paper discusses parameter manipulation abilities by way of a comparison between two controllers: the homemade instrument and the *Korg nanoKONTROL*. Mapping strategies – in an interactive music sense – are explored in detail, while the execution and capabilities of parameter control by use of the physical interface devices of the two controllers are assessed.

1. HOMEMADE REACTABLE

Using instructional information found primarily on the *reacTIVision* (*reacTIVision 1.4* nd*)* website and in the paper entitled *ReacTIVision: A computer-vision frameworks for table-based tangible interaction* (Kaltenbrunner et al. 2007), and in conjunction with original creative ideas, a homemade reacTable was designed and constructed by Masters by Research student James Herrington (see Figure 1). The instrument, with a tabletop tangible user interface, incorporates multi-touch technology, and is based on the technology of the original reacTable (Jorda et al. 2005). It can be played by a single performer, or by multiple performers.

Like the *reacTable*, this instrument incorporates a clear tabletop with a camera placed beneath, which constantly examines the table surface, tracking the nature, position and orientation of the tangibles, or objects, that are placed, and moved around, on it. The tangibles display visual symbols, called *fiducials* (see Figure 2), which are recognised by the software. Each tangible is dedicated a function for generating or manipulating/controlling a sound.

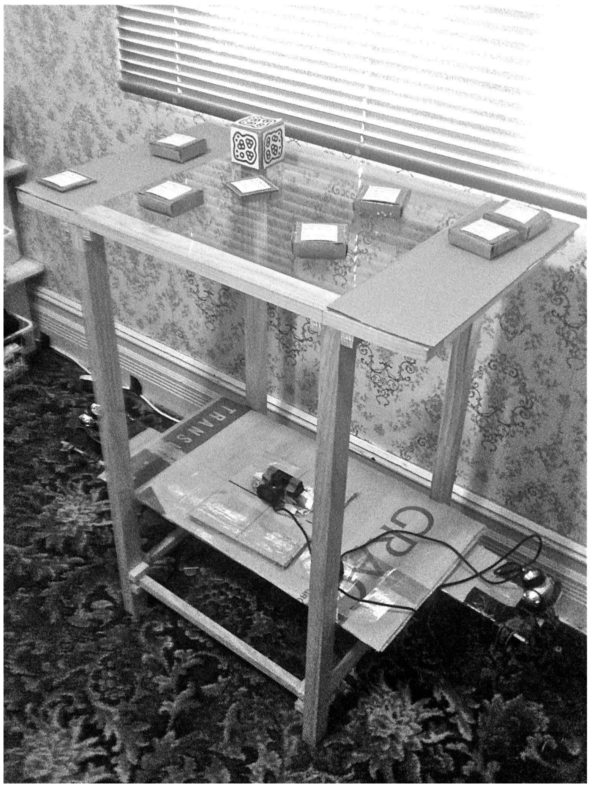

Figure 1: Homemade reacTable

Users interact by moving them around the tabletop, changing their position, their orientations, or their faces (in the case of, say, a cube object) (Jorda et al. 2005; Jorda et al. 2006).

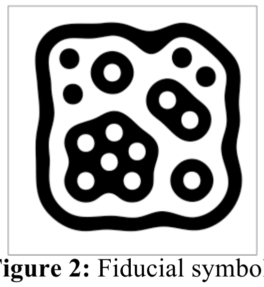

Figure 2: Fiducial symbol

This is where this instrument differs from the original *reacTable*. The vision captured by the camera is sent to the open source software *ReacTIVision*, and then to *Max/MSP*, which allows the instrument to work as a MIDI controller. This information is then sent to *Proppellerhead*'s *Reason*, where the final mapping is completed to allow note on/off events (determined by a tangible being placed and displaced in the camera's vision), along with the x-position, y-position, and orientation of each tangible assigned to manipulate different parameters of music[1].

In recent years, the availability of the previously mentioned software, and growing information on the subject, have resulted in a number of computer musicians and artists creating their own reacTables (or adaptations of them)[2]. At a tertiary level, researchers, such as the Computational Arts Research Group at the Queensland University of Technology, who developed the "Morph Table"[3] (Brown et al. 2007), are also working with the technology. While this use of the technology is utilized to develop an instrument with a set purpose of expanding certain performance and compositional techniques (i.e., morphing between musical patterns), my own homemade reacTable is set up to act as a basic MIDI controller, where the placement/displacement of any object on the table can be assigned any available note on/off function and the movement assigned to control any available parameter.

1.1 Basic physical design and build

The wooden frame structure of the homemade reacTable is based (as the name might suggest) on the shape and design of a table. The table stands 92cm high, at perfect mid-waist height. As it is intended to be performed while standing up, this gives the performer a "birds-eye" view of the tabletop, while relieving them from having to bend or sit down to move the objects around. The dimensions of the tabletop interface – clear Perspex – are 46cm (length) x 37.6cm (width). This provides the performer with quite a large area (1729.6cm^2) to move the objects around. As part of the design, on either side of the interface are two shelves (15cm x 46cm) intended for unused, or "standby", objects to rest on.

A *Sony PlayStation 3 Eye* webcam (*PlayStation Eye*. 2011) – with approx. dimensions of 84 x 67 x 57mm, and a video capture of 640 x 480 pixel – is placed 61cm directly beneath the tabletop, facing upwards in order to capture the vision of the objects being moved around. Two LED torches are placed on either side of the table, on the same level plane as the camera, but roughly 25cm to the left and right respectively of being directly underneath the tabletop interface. They are then angled to shine on the bottom side of the Perspex. This was necessary, as the visual fiducial symbols needed ample light in order to be properly recognised and sufficiently tracked when placed on the table surface. The camera is then able to constantly examine the interface, without any distracting light reflection – as the torches were strategically placed to the sides, rather than directly underneath the clear Perspex.

When it comes to the tangible objects of the homemade reacTable, they can be sorted into four categories of size. Objects can be of all shapes and sizes, and will work as long as they have an attached, and recognisable, fiducial. However, the four size groups used are as follows:

- Cube object – 7cm x 7cm x 7cm (Figure 3)
- Rectangular prism object – 7cm x 7cm x 2cm (Figure 4)
- Large flat object - 7cm x 7cm (Figure 5)
- Small flat object – 5cm x 5cm (Figure 6)

The first three listed objects – the larger objects – when placed on the tabletop interface, take up the same amount of surface area (49cm^2 – on which the fiducial is presented). The small flat object, however, takes up a lesser amount of surface area, with a smaller fiducial attached to a surface area of 25cm^2.

Figure 3: Cube object

Figure 4: Rectangular prism object

Figure 5: Large flat object

Figure 6: Small flat object

1.2 Computer software: *ReacTIVision*, *Max/MSP* and *Reason*

Regarding the computer aspect of the instrument, three software programs are used in conjunction with each other in order for vision to be captured, analysed and then interpreted into sound, or in other words, for the instrument to function. The three computer software programs, which act as the "engine room" of the instrument, are *ReacTIVision* (*reacTIVision 1.4* nd), *Max/MSP* (Cycling '74 2011) and *Reason* (Propellerhead 2011).

ReacTIVision, developed by Martin Kaltenbrunner and Ross Bencina, is the fundamental sensor

[1]Alternative software – and software combinations – can be used as opposed to *Max/MSP* and *Reason* in the set up of a homemade reacTable. This information can be found on the *reacTIVision* website.

[2]Several of these projects are documented anonymously online, including (http://www.youtube.com/watch?v=JuQo25KYELg) and (http://www.youtube.com/watch?v=15IE6d4zfME&feature=related).

[3]See (http://www.youtube.com/watch?v=nKXhfApKCms)

component of the homemade reacTable. The software is open source and can be found at the *ReacTIVision* website (*reacTIVision 1.4* nd), along with information about the internal structures and workings of the software, and instructions on how to set it up. As its function is the analysing of visual information captured by the camera placed beneath the tabletop, *ReacTIVision* does not contain any sound components. Instead, Tangible User Interface Object (TUIO) messages are sent to a TUIO-enabled client application: in the case of my instrument, this is *Max/MSP* (*reacTIVision 1.4* nd).

Max/MSP (version 5) acts as the client application in the instrument. Here, the fiducials' recognition, centre point and orientation information is processed and organised into four groups of numbers: note on/off (0 – 1), x-position (0 – 640), y-position (0 – 480) and angle (0 – 360) [The fiducials' recognition/derecognition relating to note on/off; centre point relating to x and y position; and orientation relating to angle, or rotation]. Using various techniques in *Max/MSP*, this information was organised in such a way that the zero point was located at the bottom, left hand corner of the table. For example, moving an object from left to right raises the value of the x-axis number, while moving an object from bottom to top raises the value of the y-axis number. The processing of information was also organised so that the value of the angle, or orientation, number rises when an object is rotated clockwise. These sets of numbers are then scaled to 0 – 127 in order to be sent as MIDI information to the computer software program *Reason*.

Reason (version 4) completes the process of interpreting object recognition and movement into sound generation and control. To sum up, *ReacTIVision* has analysed vision of objects and their placements, and sent this information to *Max/MSP* where it has been organised into sets of note on/off, x-position, y-position and orientation values and finally sent to *Reason*. *Reason* is where the mapping of these values to parameters of music occurs. An example would be if the y-position value of an object were assigned to the pitch shift parameter, therefore enabling the movement of this object from bottom to top of the table interface to raise the pitch of the sound produced.

2. MUSICAL PARAMETER MANIPULATION

One of the main advantages of the homemade reacTable is the number of musical parameter manipulation possibilities that can be achieved through the use of various mapping strategies. Mapping, in terms of interactive music systems, is the connection between the outputs of a gestural controller and the inputs of a sound generator. The method is typically used to link performer actions to the generation and control of musical sounds and parameters (Drummond 2009; Wanderley 2001; Winkler 1998).

2.1 Homemade reacTable mapping strategies

When it comes to the homemade reacTable, the mapping is the relationship between the movement of the tangible objects and the sounds produced. The relationships can be set up in an obvious, or not so obvious way, and ideally a balance between the two makes for a more interesting instrument. As discussed above, parameters can be mapped (or assigned) to the x-axis, y-axis and rotation of each object, while note on/off functions can be mapped to the recognition/derecognition of objects on the table interface.

Throughout this paper, specific parameter mapping assignments of the homemade reacTable have not been discussed, other than in examples, because, discussing them in detail here (i.e., each object's function) is not important. It is only important to know that the placement of objects on and off the table surface can generate any accessible sound and activate any accessible effect, and that the x-axis, y-axis and angle movement of the object can manipulate any accessible parameter of the music.

There are four main mapping strategies that can be used in interactive music systems: *one-to-one*, which is the direct connection of an output to an input; *one-to-many*, which is the connection of a single output to multiple inputs; *many-to-one*, which is the connection of two or more outputs to control one input; and *many-to-many*, which is a combination of the different mapping types (Drummond 2009; Miranda and Wanderley 2006). This is certainly apparent in the homemade reacTable, where the output relates to one type of an object's movement (i.e., x-axis or y-axis or angle movement), and the input relates to any desired parameter of music to be manipulated.

A noteworthy feature of the objects is that, because of the way the instrument is set up – specifically the *Max/MSP* element – objects can continually rotate. That is so that the value assigned to the orientation of the object resets to 0 after a full rotation. This enables control of the parameter so that the value can switch straight from MIDI CC 127 to MIDI CC 0, or vice versa. The x-position, y-position and orientation of each object can be used to manipulate parameters of the same "parent" effect (e.g., x = reverb dry/wet amount, y = reverb decay, and angle = HF damp), or alternatively be used to manipulate completely different parameters (e.g., x = reverb dry/wet amount, y = pitch shift, and angle = panning).

2.2 Comparison

The parameter manipulation abilities of the homemade reacTable can be contrasted with the manipulation abilities of a commercial MIDI control device, the *Korg nanoKONTROL* (*Korg* 2011). The choice in comparing the *nanoKontrol* is because, as a general MIDI controller that sends MIDI information to MIDI-enabled devices, it incorporates the basic and universal note on/off functions, and also interface-controls that can access every MIDI CC value. Although new and experimental MIDI interfaces are being developed all the time, the most common physical MIDI controllers contain pads, or keys, for note on/off functions, and knobs/faders for continuous signal control. This USB bus powered device offers nine faders, nine knobs and 18 switches, with four programmable scenes, along with a full transport section (controlling functions such as *start, stop, loop,* or *record* on the DAW software). It is a small controller with dimensions 320(W) x 82(D) x 29.5(H) mm, and a weight of 29g.

It is important to note that the assessment will only take into account the knobs and faders of the *nanoKONTROL*, which can access every MIDI CC value (0 – 127). Although they can transmit MIDI CC messages, the 18 switches on the device can only access, or trigger, two values. Attack and decay times can be assigned to the switches, and therefore access every value if set up correctly; however, this is a set function and the MIDI CC values cannot be continually controlled. When it comes to the homemade reacTable, the x, y and orientation position of each object (or more specifically, fiducial) can access every MIDI CC value.

In the comparison below, the parameter manipulating devices will be referred to as Physical Interface Controllers (PICs). In the case of the homemade reacTable, this term will refer to the tangible objects with attached fiducial symbols. In the case of the *nanoKONTROL*, this will refer to the knobs and faders of the device. The devices are compared below on four dimensions, in the form of questions relating to the execution and capabilities of parameter control. The two controllers are assessed as if performed as a solo instrument by a solo performer.

How many potential PICs can be used/assigned to manipulate parameters of the music?

<u>Homemade reacTable:</u>
In the setup of the homemade reacTable, the default 'amoeba' fiducial set[4] is used, which includes 216 distinct fiducials. Fiducial IDs 0 – 107 being a black image on white, while fiducial IDs 108 – 216 are the inverse, with the same images reversed (i.e., now a white image on black). Each of these fiducials can be tracked and therefore can be assigned to manipulate different parameters of music. This means that 216 potential PICs can be set up for use in the one instrument (*reacTIVision 1.4* nd).

<u>*nanoKONTROL:*</u>
The *nanoKONTROL*'s interface consists of nine knobs and nine faders with four programmable scenes. As each scenes memory allows the same settings to be retained, this results in 36 knobs and 36 faders that can be assigned to manipulate different parameters of music. Therefore, 72 potential PICs can be set up for use with the controller.

[4]Alternative sets of fiducials are also available to be used and are available from the *reacTIVision* website.

How many PICs can be used/played/controlled simultaneously?

This question requires two distinct responses. The first relates to the idea that multiple PICs can be accessed and easily moved between when interacting with the instrument, and also without the parameter assignment being changed or replaced mid-performance. The second response relates to the amount of PICs that can be controlled simultaneously under the physical limitation of the human performer.

<u>Homemade reacTable:</u>
Although 216 fiducials can be assigned to manipulate different parameters of music, the objects they are attached to cannot possibly fit on the tabletop interface of the homemade reacTable all at the one time.

In a practical experiment, multiple performances of the instrument using a different amount of objects were trialled, in ways utilizing alternative mapping strategies for each set of objects. Only larger objects were used, those with a surface area of $49cm^2$ – which is placed on the interface. The kind of music being performed is not specified, nor the different parameters of music used in the mapping to the objects' movement, for this information is not relevant, as the experiment is to determine the *number* of objects that can be can be interacted with on the interface in a comfortable manner.

Firstly, parameters were only assigned to the objects' x-axis and orientation (i.e., no parameters were assigned to the y-axis of each object). Set up this way, the objects were lined up from top to bottom so that the full x-axis range of values could be realised (i.e., each object could be moved from the left most point of the table [MIDI CC 0] to the right most point of the table [MIDI CC 127]) without clashing with each other. Using this mapping strategy, four objects appeared to be an adequate number. Taking into account the size of the table and the size of the objects, there was comfortable room between the objects so they could all be rotated without interference while on the same x-axis point.

Secondly, the parameter-to-object mapping was set in the same way as mentioned above; however, only assigning parameters to the y-axes and orientations of the objects (rather than the x-axes and orientations). Once again, four objects was found to also be an adequate number. That is to say, when lining the objects from left to right, each object could access every y-axis value without clashing with other objects, and in fact more room – or empty space – was available between the objects. This is because the width of the tabletop interface is longer than the height.

Next, parameters were only assigned to the objects' orientation. Set up this way, the objects were lined up from bottom to top as well as from left to right in a grid fashion, with enough space between them so that the full rotation range of values could be accessed without clashing with each other. Using this strategy, 16 objects were found to be a satisfactory amount of objects on the interface. That is to say, the interface accommodated four rows of four objects.

Finally, parameters of music were assigned to the objects' x-axis, y-axis and orientation. With this mapping strategy employed, it is hard to give an exact number of how many objects allows for satisfactory performance. In a way, it depends on what type of piece is being performed. For example, an experimental free improvisation piece, where objects are moved around at free will, would accommodate more objects than a piece where one or two objects have to be moved at certain times to certain positions without interrupting the placement of other objects. If the piece is in fact structured in this way, the performer would need to pre-determine which objects need to move along entire x and y-axes, and arrange the objects accordingly when first placing them on the table. In saying this, nine objects (three rows of three) were found to be an adequate number where objects can move around freely enough.

When it comes to the number of PICs that can be used simultaneously (i.e., how many on the interface at the one time), it really depends on a variety of factors, such as the mapping strategies employed. Another factor, not discussed thus far, is the size of the objects. The examples above utilized objects with a surface area of 49cm^2. As mentioned, smaller sized objects can be used, which would enable more simultaneous PICs on the interface at the one time. Combinations of larger and smaller objects can also be used. It is also worth noting that when the performer needs to change or replace certain PICs, and thus the parameters to be manipulated, one object at a time can be replaced on the interface.

The second response to the original question involves the human performer's limitations in the physical controlling of the objects. Traditionally, or rather ideally, two objects can be controlled simultaneously by a solo human performer, that is, one in each hand[5]. We say 'ideally' because more than two objects *can* physically be controlled, however, when doing so, restrictions arise. For example, the performer could use his or her nose or teeth, like a modern day Jimi Hendrix, to control a third object, however – when compared to controlling objects with one's hands – this can hardly be achieved efficiently, as it would be awkward. A second example occurs when the performer moves two or more objects with the one hand. Once again, to avoid being extremely awkward, this can only take place if the two or more objects are to be executing the same control, that is, moving the multiple objects up, down, left or right simultaneously. Rotating the objects with the one hand would be difficult without also altering the x and y positions of the object.

<u>*nanoKONTROL:*</u>

As realised in the previous question, 72 PICs can be set up from nine knobs and nine faders on the physical interface of the *nanoKONTROL*, however, only these nine knobs and nine faders (i.e., 18 PICs) can be used without altering the scene, and thus replacing the assigned parameter set of one scene with a completely different set of another scene. The act of switching between scenes directly opposes the idea of being able to easily access and move between parameters to manipulate. For example, it would not be possible to manipulate the parameter assigned to *knob 1* of the first scene and the parameter assigned to *knob 2* of the second scene simultaneously. This applies to any two separate parameters assigned to PICs on contradicting scenes. Therefore, only the 18 (physical) PICs can be controlled "simultaneously" when relating to the first response of the original question. Unlike the homemade reacTable, where one PIC can be changed or replaced at a time, the *nanoKONTROL* can only move between scenes, and therefore 18 PICs (or rather the parameters assigned to them) can only be changed or replaced by 18 PICs at a time.

As considered with the homemade reacTable, to answer the second part of the question involves discussing the human performance confinements in the physical controlling of the – in this case – knobs and faders. Like the homemade reacTable, the ideal number of faders and/or knobs to be controlled simultaneously is two – one with each hand[6]. The same "slight exceptions" examples also relate. That is to say, a performer *could* use his or her nose or teeth (or any other part of the body) to control a PIC, however, it would be awkward; while two or more faders (not so much knobs) can be controlled simultaneously, with enough amount of efficiency, with the one hand, although, only when performing the same control – in this case, being moved up or down.

How many parameters of music can be independently manipulated using the one PIC?

<u>*Homemade reacTable:*</u>

When it comes to the homemade reacTable, three parameters can be independently controlled using the one PIC. That is to say, a parameter of music each can

[5]Multiple performers result in more objects being able to be controlled simultaneously. For example, two performers on the one instrument can control four objects, three can control six objects, and four can control eight objects, with each performer controlling the standard two objects. More performers can be added, but because of the size of the instrument, however, the space would become more and more cramped when consisting of more than four players. The square design of the instrument also neatly accommodates four players, with one performer on each side.

[6]Once again, multiple performers result in more faders and knobs being able to be controlled simultaneously, with each performer controlling two PICs. However, as opposed to the homemade reacTable, at a much smaller physical size, it would be extremely cramped with any more than two performers.

be assigned to the movement of the object on the x-axis of the tabletop interface, the movement on the y-axis, and the angle – or orientation – of the object. Each parameter can be manipulated separately. For example, an object can be moved from left to right on the table, controlling the parameter assigned to the x-axis, while maintaining the values, or settings, of the parameters assigned to the y-axis and angle of the object.

Alternatively, the parameters can be manipulated simultaneously at an independent rate. For example, moving the object in an oval-shaped manner, while continuously rotating the object. Each parameter is being manipulated at a different rate, while changes in the shape of the movement add to the independent parameter manipulation possibilities. Furthermore, changing the rotating speed of the object will affect the manipulation rate of the parameter assigned to the angle of the object independently of (i.e., without affecting) the parameters assigned to the x and y-axes[7].

<u>*nanoKONTROL:*</u>
Only one parameter of music can be independently controlled using only one PIC of the *nanoKONTROL*. Multiple parameters can be assigned to the one fader (or knob alternatively), for example, however, when the fader is controlled (i.e., raised and lowered), the value of each parameter is manipulated at the same rate.

Using various external software, one can alter the nature of how each of the parameters assigned to the one PIC is manipulated. Using a fader once again as an example, the minimum and maximum values can be reversed, so as the fader is physically raised, the value of the parameter is lowered. The minimum and maximum vales can also be restricted to a certain range, so as a fader is raised from the minimum position to maximum position on the physical instrument, the MIDI CC value of the parameter would raise, for example, from 10 – 80 (or whatever range the user has assigned) as opposed to 0 – 127[8]. However, even if both of these examples were allocated to two different parameters assigned to the one PIC, along with another parameter being manipulated in the traditional sense (minimum position to maximum position on the physical interface equates to 0 – 127 parameter value), the three parameters would all be manipulated at the same exponential rate when controlled by the one fader, that is, not independently of one another[9].

Are there any placement restrictions of the PICs?

<u>*Homemade reacTable:*</u>
Objects are placed and moved around the tabletop interface of the instrument in order to achieve their assigned parameter manipulation functions. In saying this, it is not physically possible for two objects to be in the same xy position on the table surface. Depending on the parameter assignments of each object, this means that certain combinations of audio effects are unachievable. Because of this limitation, a good mapping strategy technique would be to assign music parameters to the movement of the objects' orientation and only *one* of their axes (x or y). This is because an object can be rotated on the table surface without affecting its xy location, and therefore, without clashing with other objects.

Experimenting with the larger objects ($49cm^2$) on the homemade reacTable, it was found that two objects with the same y-position, and side-by-side as close as possible to the same x-position, could not access any values within MIDI CC 20 of each other on the x-axis. Two objects with the same x-position, as close as possible to the same y-position could not access values within MIDI CC 28 on the y-axis. Once again, this is due to the fact that the width of the interface is longer than the height. Using the smaller objects ($25cm^2$), two with the same y-position, could not access within MIDI CC 15 on the x-axis, while two with the same x-position could not access within MIDI CC 20 on the y-axis[10].

<u>*nanoKONTROL:*</u>
Unlike the homemade reacTable, where parameters are manipulated by moving each object around the one surface plane, the knobs and faders of the *nanoKONTROL* are allocated their own space. Because of this, there are no placement limitations of the PICs.

An example of the difference between the two controllers under assessment (the homemade reacTable and the *nanoKONTROL*) would be as follows:

[Using only x and y possibilities, and ignoring the rotation parameter control function for now] *Object 1* of the homemade reacTable controls the reverberation of the entire sound through the use of a reverb unit in *Reason*. The parameter assigned to the movement of the x-axis is the amount of decay, and the parameter

---

[7]Multiple fiducials can be placed on the one object, such as a larger object with four fiducials displayed on the one face, and played in a way where one hand is controlling the object. This raises the amount of multiple parameters controlled by one PIC, however, in doing so disregards the idea of independent parameter manipulation between the parameters assigned to the two or more fiducials on the one object. That is to say, however the object is moved around the tabletop interface, all fiducials on the object move in the same way.

[8]Using the *Korg Kontrol Editor* (*nanoKONTROL* 2011), one can edit settings on the *nanoKONTROL* itself (i.e., without using external software) to modify how each PIC manipulates its assigned parameters (as discussed above), although, three alternative ways cannot be designated to the one PIC using this method. That method can only be achieved by using external software, such as *Ableton Live* (*Ableton* 2010).

[9]*Korg* has released a controller, different to the *nanoKONTROL*, called the *nanoPAD* (*Korg* 2011) with an xy pad (where x and y parameters can be manipulated independently) controlled by finger touch. In this case, the user's finger would act as the PIC.

[10]The larger the interface surface area is, the smaller these number values would become

	Homemade reacTable	Korg nanoKONTROL
No. of potential PICs	216	72
No. of PICs that can be controlled simultaneously	*In the same space without being changed/replaced*: Larger Objects (49cm^2 surface area): X and ANG controlling parameters: 4 Y and ANG controlling parameters: 4 ANG controlling parameters: 16 X, Y and ANG controlling parameters: 9 *Under human limitations*: 2	*In the same space without being changed/replaced*: 18: 9 faders, 9 knobs on 1 scene *Under human limitations*: 2
No. of parameters that can be independently manipulated using one PIC	3	1
Placement restrictions	2 objects can not be in the same xy position on the interface	No restrictions

Table 1: Comparison of homemade reacTable and *Korg nanoKONTROL* across 4 dimensions

assigned to the movement of the y-axis is the dry/wet amount. Meanwhile, *object 2* controls the equalisation of the entire sound through the use of a Parametric EQ unit. The parameter assigned to the movement of the x-axis is the frequency value, and the parameter assigned to the movement of the y-axis is the gain value. If the performer wishes to achieve the audio effect of a reverberation with a decay amount of MIDI CC 50 and a dry/wet amount of MIDI CC 30, simultaneously with the equalisation emphasising the frequency at MIDI CC 50 at a gain of MIDI CC 30, this unfortunately would not be possible. This is because, in the case of both object placements, x = 50 and y = 30, and two objects cannot be in the same xy position. To overcome this problem, various mapping strategies can be employed, as discussed previously. One approach would be to switch the assigned x and y parameters of one of the objects, resulting in one object requiring x = 50 and y = 30, and the second object requiring x = 30 and y = 50 to achieve the desired audio combination effect. This method would not work, however, if all four parameters (the parameters assigned to x and y movement of both objects) required the same MIDI CC value, that is, if both objects required the values x = 50 and y = 50.

In the case of the *nanoKONTROL*, using the same parameters and intended values, the audio effect combination *can* be achieved. Using four faders (or knobs alternatively) *fader 1*, assigned to the reverb decay amount, can achieve the MIDI CC value of 50; *fader 2*, assigned to the reverb dry/wet amount can achieve the MIDI CC value of 30; *fader 3*, assigned to the frequency of the EQ, can achieve the MIDI CC value of 50; while *fader 4*, assigned to the gain of the EQ, can achieve the MIDI CC value of 30. These values can be achieved simultaneously, unlike on the homemade reacTable, however, two additional PICs must be used.

The potential of each controller in relation to the four questions is summarised in Table 1 above.

2.2.1 Other things to consider

The above assessment has compared the parameter manipulation abilities of two controllers – the homemade reacTable and the *Korg nanoKONTROL*. Although, if the two were to be compared as the better overall, or more useful, controller, various other aspects would need to be considered. This may include the lag or delay between the physical movement of a PIC and the assigned parameter value. The CPU usage of each would also need to be assessed, as would the restrictions due to size, and the ease of portability.

3. FUTURE WORK

Future work on parameter manipulation utilizing this type of technology could include setting up the homemade reacTable instrument in a way so that the distance between two objects acts as another value that can be calculated and assigned to an additional musical parameter to be controlled. This way, both objects would have to be present on the interface in order for the parameter to be altered. If three parameters each were additionally mapped to the x-axis, y-axis and orientation of the two objects, this would enable the performer to control seven parameters of music independently of each other using only two objects. For many pieces of music, this would be all the control the performer needs.

Although the lag and delay due to the quality of the webcam was not discussed throughout the paper, future work may also involve the use of a High Definition camera to track the fiducials more effectively.

For his own artistic endeavors, James Herrington intends to further this research to develop alternative ways to perform and compose contemporary electronic music, and use these extensive manipulation possibilities to advance the homemade reacTable as a DJ instrument. He is also currently working with the instrument as a component in an integrated Dubstep Performance environment for his Masters by Research project.

4. CONCLUSION

Advances in music technology in conjunction with increasing accessibility and affordability have contributed to progress in the area of musical parameter manipulation in live electronic computer music. This is made apparent by comparing a homemade electronic instrument with a commercial controller, and showing that in many ways the homemade instrument possesses superior parameter manipulation abilities. Apart from the computer itself, the sum of components of the homemade instrument – including open source, and relatively inexpensive software – come at a reasonable price. The PICs of the homemade reacTable as tangible, freely moving objects generate the novel characteristics, and enable the excellent range of parameter manipulation, of the instrument. A greater scope of versatility and control of the musical output is produced, that, by comparison, can be restrictive when it comes to other electronic devices and controllers. By no means have all possibilities in parameter manipulation utilizing this technology been explored, which leaves the door open for further investigation and exciting advances.

5. REFERENCES

Ableton. 2010. [cited March 23 2011]. Available from http://www.ableton.com/.

Brown, A. R, R. W. Wooler, and K. Thomas. 2007. "The Morph Table: A collaborative interface for musical interaction." Paper read at *Australian Computer Music Conference* at Canberra, Australia.

Drummond, J. 2009. "Understanding interactive systems." *Organised Sound* 14 (2):124-133.

Jorda, S, M Kaltenbrunner, G Geiger, and M Alonso. 2006. "The reacTable: A tangible tabletop musical instrument and collaborative workbench."

Jorda, S, M Kaltenbrunner, G Geiger, and R Bencina. 2005. "The reactable*." Paper read at *International Computer Music Conference*, at Barcelona, Spain.

Kaltenbrunner, M. 2009. "ReacTIVision and TUIO: A tangible tabletop toolkit." Paper read at *ACM International Confeence on interactive tabletops and surfaces* at Banff, Alberta, Canada.

Kaltenbrunner, M., and R. Bencina. 2007. "ReacTIVision: A computer-vision frameworks for table-based tangible interaction." In *Proceedings of the First International Conference of Tangible and Embedded Interaction (TEI'07)*. Baton Rouge, LA.

Korg: nanoSERIES: Slimline USB MIDI Controllers 2011. [cited March 21 2011]. Available from http://korg.com/Product.aspx?pd=415.

Max MSP 5. Cycling '74. 2011. [cited March 20 2011]. Available from http://cycling74.com/

Miranda, E. R., and M. M. Wanderley. 2006. *New digital musical instruments: Control and interaction beyond the keyboard*. Vol. 21, *The Computer Music and Digital Audio Series*. Middleton, WI: A-R Editions.

nanoKONTROL: Slim-Line USB Controller: Owner's Manual. 2008. [cited March 21 2011]. Available from http://www.korg.co.uk/downloads/nano/support/nanoKONTROL_OM_E2.pdf.

PlayStation Eye. 2011. [cited March 19 2011]. Available from http://au.playstation.com/ps3/peripherals/detail/item78696/PlayStation-Eye/.

reacTIVision 1.4: a toolkit for tangible multi-touch surfaces. nd. [cited 15 March 2011]. Available from http://reactivision.sourceforge.net/.

Reason 4. Propellerhead. 2011. [cited March 20 2011]. Available from http://www.propellerheads.se/

Wanderley, M. 2001. "Gestural control of music." Paper read at *International Workshop - Human Supervision and Control in Enginerring and Music*, at Kassel, Germany.

Winkler, T. 1998. *Composing interactive music: Techniques and ideas using Max*. Cambridge, MA: The MIT Press.

Appendix 2

Performance 1 score

SCORE: Basic structure

Performance 1

SECTION 1

- Drums enter

Showcase:	X-pos.	*LP Filter position*
	Y-pos.	*Distortion amount*
	Cube sides	Different loops

DRUMS hold steady – sparse loop – no *LP Filter position* or *Distortion amount* changes

- Synth enter

Showcase: *Timbral sudden changes*
Amplitude/timbre

SYNTH plays sparse, simple riff – no *Timbral sudden changes* or *Amplitude/timbre* changes
DRUMS remain the same

- Vocals enter

Showcase: *Frequency*
Amplitude

SECTION 2

Synth [affecting]
1) Long steady control, and 2) no thought, just playing-own-layer mindset

Vocals

Timbral sudden changes:	*Tremolo on/off*
Amplitude/timbre:	*Reverb amount*

Drums

Timbral sudden changes:	*ADSR on specific hit*
Amplitude/timbre:	*Amplitude*

VOCALS [being affected]
Long steady notes – no rapid changes in *Frequency* or *Amplitude*

DRUMS [being affected]
One loop – dry *LP Filter position* and *Distortion amount*

Vocals [affecting]
1) Long steady control, and 2) no thought, just playing-own-layer mindset

Synth

Frequency: *HP Filter position*
Amplitude: *Vibrato amount*

Drums

Frequency: *Reverb amount*
Amplitude: *Frequency of hit*

SYNTH [being affected]
Long steady notes – no rapid changes in *Timbral sudden changes* or *Amplitude/timbre*

DRUMS [being affected]
One loop – dry *LP Filter position* and *Distortion amount*

Drums [affecting]
1) Long steady control, and 2) no thought, just playing-own-layer mindset

Synth

Distortion amount: *Amplitude*
LP filter position: *Reverb amount*

Vocals

Distortion amount: *BP filter position*
LP filter position: *Distortion amount*

SYNTH [being affected]
Long steady notes – no rapid changes in *Timbral sudden changes* or *Amplitude/timbre*

VOCALS [being affected]
Long steady notes – no rapid changes in *Frequency* or *Amplitude*

SECTION 3

Jam like a band
Have no controlling of other layers' parameters in mind

OUTRO